ROBUST DECENTRALIZED LQ CONTROLLER DESIGN FOR ROBOT ARM MANIPULATORS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by
HARI BABU

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
APRIL, 1989

4/4/89

CERTIFICATE

It is certified that the thesis entitled ''Robust Decentralized LQ Controller Design for Robot Arm Manipulators'' has been carried out by Hari Babu under my supervision and that this work has not been submitted elsewhere for a degree.

March, 1989.

(K.E. Holé)
Professor
Dept. of Electrical Engg.
I.I.T. KANPUR
INDIA.

ACKNOWLEDGEMENT

When I think of my toils over the thesis, I feel that without the much needed guidance and encouragement of Prof. K.E. Hole, it would not have been possible for me to complete the work. At this moment I sincerely thank him for all his guidance, encouragements and discussions.

I am also thankful to Dr. R. Hradaynath, former Director, Dr. O.P. Nizhawan, Director and Mr. J.A.R. Krishnamoorty, Scientist 'E', Instruments Research and Development Estt., Dehradun, without whose initiative and interest, it would not have been possible for me to come to IIT KANPUR for this course.

I am indebted to my wife who was a source of constant encouragement particularly at times when I used to be depressed when not getting results.

I am thankful to all my friends particularly M/s C. Seshadri, A.K. Dalmia, R.V. Jawdekar and M. Raghunandan, discussions with whom contributed directly or indirectly in the accomplishment of this thesis.

Last, but not the least, I thank Mr. L.S. Bajpai for neat typing of the manuscript.

Hari Babu
(HARI BABU)

IIT KANPUR
MARCH, 1989.

CONTENTS

Table of Symbols

A	:	State matrix
A_c	:	Interconnection matrix
A_D	:	Diagonal state matrix after interconnection terms have been removed
A_i	:	Homogeneous 4x4 co-ordinate transformation matrix relating i^{th} co-ordinate frame to $i-1^{th}$ co-ordinate frame
A_p	:	State matrix of the plant
B	:	Input matrix
B_D	:	Diagonal input matrix after interconnections have been removed.
$\underline{C}$	:	Gravity loading vector
D	:	nxn symmetric p.d. inertia matrix
D^o	:	Inertia-force/torque - acceleration sensitivity matrix evaluated about the nominal trajectory
$\underline{f}$	:	Input force/torque vector
$\underline{f}_o$	:	Initial value of input force/torque vector
$\underline{g}$	:	Gravity vector
$\underline{h}$	:	Centrifugal, coriolis and gyroscopic force/torque vector
I	:	Identity matrix
	:	Co-ordinates of the c-o-g of the i^{th} link wrt i^{th} co-ordinate frame
	:	Performance index
	:	Pseudo inertia tensor of i^{th} link

L : Feedback gain matrix

m_i : Mass of i^{th} link

n : Number of joints

P^o : Force/torque - position sensitivity matrix evaluated about the nominal trajectory

Q : State weighting matrix

$\underline{q}$: Joint variable vector

$\underline{\dot{q}}$: Joint velocity vector

$\underline{\ddot{q}}$: Joint acceleration vector

$\underline{u}$: Input vector

V^o : Centrifugal, coriolis and gyroscopic force/torque - velocity sensitivity matrix evaluated about a nominal trajectory

W_i : Homogeneous 4x4 co-ordinate transformation matrix relating i^{th} co-ordinate frame to base co-ordinate frame.

$\underline{x}$: State vector

$\underline{x}_o$: Initial value of state vector

α : Margin of stability

$\underline{\delta f}$: Small deviation in the nominal value of $\underline{f}$

$\underline{\delta q}$: Small deviation in the nominal value of $\underline{q}$

$\underline{\delta \ddot{q}}$: Small deviation in the nominal value of $\underline{\dot{q}}$

$\underline{\delta \dot{q}}$: Small deviation in the nominal value of $\underline{\ddot{q}}$

Σ : Input weighting matrix.

ABSTRACT

The problem of designing a robust decentralized linear quadratic controller of a robot arm manipulator has been tackled in this thesis. Recursive Lagrange-Euler equation of robot arm motion has been linearized around a trajectory and linear error models have been obtained. A decentralized controller has been designed for a PUMA-560 robot and its performance has been evaluated for various load conditions. Tracking of the trajectory has been found satisfactory under these load conditions. The important conclusion of the thesis is that only one controller is used for the whole trajectory and there is no need to calculate the controllers on line.

CHAPTER 1

INTRODUCTION

1.1 ROBOT MANIPULATOR

An industrial robot is a general purpose, computer controlled manipulator consisting of several rigid links connected in series by revolute or prismatic joints. One end of this chain is attached to a supporting base, while the other end is free and equipped with a tool to manipulate objects or perform assembly task.

A robot manipulator consistsof an arm sub-assembly and a wrist sub-assembly plus a tool. The arm sub-assembly generally can move with three degrees of freedom. The combination of the movements of arm sub-assembly positions the wrist unit at the desired location. The wrist sub-assembly unit usually consists of three rotary motions and orients the tool according to the configuration of the work piece.

1.1.2 Robot Control Problem

The control problem of a robot manipulator can be conveniently divided into two coherent sub-problems [10] :

1) Motion (or trajectory)planning sub-problem, and
2) Motion control sub-problem.

The trajectory planner interpolates and/or approximates the desired path by a class of polynomial functions and

generates a sequence of time based control set points for the control of manipulator from the initial location to the destination location. The trajectory planning can be done either in joint variable space or in the cartesian space. (For a robot with revolute joints, the joint variables are the angles between the links and for a robot with prismatic joints, the joint variables are the distances between the joints). For joint variable space planning, the time history of all joint variables and their first two time derivatives are planned to describe the desired motion of the manipulator. For cartesian space planning, the time history of manipulator hand's position, velocity and acceleration are planned.

The motion control problem consists of two parts :

1) Obtaining dynamic models of the manipulator, and
2) Using these models to determine control laws or strategies to achieve the desired system response and performance.

Robot arm dynamics deals with the mathematical formulations of the equation of motion of the robot arm. The mathematical formulation of robot arm motion is necessary for computer simulation of the robot arm, design of suitable control equation, and evaluation of the kinematic design and structure of the robot arm. The last use of dynamic equation of robot motion is not included in the present work. In the present work, the

dynamic equation has been evaluated for obtaining a nominal control for the robot joints and for a computer simulation of the arm.

As it turns out, the equations of robot arm motion are highly nonlinear and the established design methods for linear systems cannot be applied directly.

Several authors have tried to design the controller for the robot manipulator based on various techniques. A great majority of them have developed the controller in joint-variable space by nonlinear compensation of the coupling forces among the various joints[10]. But this method of designing the controller has a serious draw back in the sense that the non-linearities may not be exactly cancelled by the non-linear compensators which may result in poor closed loop stability and performance [3]. Hence, as far as possible, the nonlinear compensation should be avoided.

An alternative strategy to design a controller could be by linearizing the dynamic equation of motion around the nominal trajectory and then designing a robust controller for this linearized dynamic equation.

The controller design consists of two steps,

1) Nominal control design, and
2) Feedback control design.

1.2 CONTROL STRATEGY

In the present work, the following control strategy has been formulated. A nominal control is designed that propels the robot arm to the vicinity of the desired trajectory point. Near the trajectory points, the nonlinear equation of motion is linearized for the error in joint variables and linearized models obtained. An LQ regulator is designed for these error models that retains the optimality of the closed-loop system for all the points on the trajectory. This controller has been implemented and performance of a PUMA-560 robot has been evaluated. The trajectory tracking has been found satisfactory.

1.3 BREAK-DOWN OF CHAPTERS

In Chapter 2, various methods of evaluation of robot arm dynamics have been discussed briefly with their relative merits and demerits. A special mention of recursive Lagrange-Euler technique has been made. A method is given, in this chapter, to obtain the nominal torque for a given trajectory point in joint variables. A linear error model of the robot has been obtained using recursive Lagrange-Euler equation and algorithm to obtain the nominal torque and linear model is given.

Chapter 3 discusses the methods of robot control. In the present work an attempt was made to design a decentralized controller consisting of a nominal control and a feedback control to bring the tracking error as close to zero as possible. The decentralized controller was designed based on a technique developed in [13]. This chapter briefly reviews this technique.

The decentralized controller designed in [13] guarantees the robustness of the plant against the variations in plant parameters.

The decentralized feedback controller for the robot has been designed based on this technique for a PUMA-560 robot and implemented. This chapter presents the evaluation results of the controller for various load conditions.

Chapter 4 concludes the thesis. In this chapter, the observations have been made regarding the controller and its performance.

CHAPTER 2

EVALUATION OF ROBOT ARM DYNAMICS

2.1 INTRODUCTION

Robot arm dynamics deals with the mathematical formulation of the equation of motion of the robot arm. The dynamic equations of motion are a set of mathematical equations describing the dynamical behaviour of the manipulator. Such equations of motion are useful for computer simulation of robot arm motion and design of a suitable controller for the robot arm. Actual dynamic model of a robot arm can be obtained from physical laws such as the laws of Newtonian and Lagrangian mechanics. This leads to the development of dynamic equation of motion for various articulated joints of the manipulator in terms of specified geometric and inertial parameters of the links. Conventional approaches like Lagrange-Euler (LE) and the Newton-Euler (NE) formulations are then applied to systematically develop the actual robot arm motion equations.

As it turns out, these equations of motion are highly coupled and nonlinear. Therefore, the state equations derived from these highly nonlinear coupled dynamic equations are also nonlinear. A convenient way of representing the dynamic equation of motion of robot arm could be

$$\dot{\underline{x}} = \underline{g}\ (\underline{x},\ \underline{f}, \underline{x}_o,\ \underline{f}_o)$$

where

$\underline{x}$: State vector

$\underline{f}$: Vector of applied forces/torques

$\underline{x}_o$: Initial value of state vector

$\underline{f}_o$: Initial value of applied force/torque vector.

Although a fairly large amount of research has been devoted to the design of controllers for nonlinear systems, the design procedures are not yet as standard as those for linear systems. Therefore, the emphasis is either on global linearization of the original nonlinear system or linearizing it in the operating region via a suitable transformation. In the present work, the nominal torque/force, the torque/force that propels the joints to the sufficiently close neighbourhood of the desired trajectory, is calculated on the basis of the nonlinear dynamic equations and the feedback torque to reduce the tracking error is calculated on the basis of a linearized model (dynamic equation linearized around a trajectory).

The chapter has been organized as follows. Section 2.2 deals with various methods of calculating nominal torque alongwith their merits and demerits from the point of views of number of operations required to be performed and design of a controller. Section 2.3 gives current approaches in linearization of nonlinear dynamic equation of robot arm motion. Section 2.4 gives an algorithm

for calculating the nominal torque and linearized model about a trajectory. This algorithm has been applied to a practical example of a PUMA-560 robot and nominal torque and linearized models are obtained. It may be noted that the values of nominal torques obtained for the joint angles, velocities and acceleration as obtained here are the same as those reported in [2].

2.2 NOMINAL CONTROL

In order to move the end effector of a robot manipulator from an initial position/orientation to a final position/orientation along a specified trajectory, the controller is required to produce a nominal control torque/force which will move the end effector along a given trajectory. For this purpose as also for designing a control system to achieve satisfactory tracking, it is necessary to derive the dynamic equation of motion describing the dynamic behaviour of robotic manipulator. Several methods of deriving the dynamic equation of a robot manipulator such as Lagrange-Euler (LE) method, Newton-Euler (NE) method generalized D'Alembert (GD) equation etc. have been discussed in the literature.

All these methods, except Newton-Euler method, need very large amount of time for computer evaluation of dynamic equation. A comparision of the time required by various methods has been given in [9], [10], from which

it appears that the Newton-Euler method is best suited for computation of nominal torque as the number of multiplications and additions is proportional to the degree of freedom (n) of the robotic manipulator as opposed to n^4 and n^3 in Lagrange-Euler and generalized D'Alembert methods [10] respectively.

From the point of view of only calculating the nominal torque, one is normally governed by the following considerations while selecting a method of computation of nominal torque. These are :

1) Method should be easy to formulate.
2) Method should be easily converted to computer algorithm, and
3) It should lead to efficient numerical evaluation.

This might prompt one to use the NE method. However it is difficult for it to be applied to the design of control system as this method does not lead to a set of closed and explicit form differential or state equations required for designing an appropriate controller [5]. From this point of view, the Lagrange-Euler equations of motion are in the most perfect form for the design of control systems [5].

Several efficient algorithms have been proposed e.g. [5], [9], for evaluating the dynamic equations which reduce the number of computations required in LE method and thus bringing this method roughly at par with the NE

equation from the point of view of number of multiplications and additions required to calculate the nominal forces/torques.

The Lagrange-Euler equation of motion of robot manipulator can be written as

$$D(\underline{q})\, \ddot{\underline{q}} + \underline{h}(\underline{q}, \dot{\underline{q}}) + \underline{c}(\underline{q}) = \underline{f} \qquad (2.2.1)$$

where,

$\underline{q}$: An nx1 joint variable vector = $[q_1, q_2, \ldots q_n]^T$

$\dot{\underline{q}}$: An nx1 joint velocity vector = $[\dot{q}_1, \dot{q}_2, \ldots \dot{q}_n]^T$

$\ddot{\underline{q}}$: An nx1 joint acceleration vector = $[\ddot{q}_1, \ddot{q}_2, \ldots, \ddot{q}_n]^T$

D : nxn symmetric positive definite inertia matrix.

$\underline{h}$: An nx1 coriolis , centrifugal and gyroscopic force/torque. vector = $[h_1, h_2, \ldots, h_n]^T$

$\underline{c}$: An nx1 Gravity loading force/torque vector $[c_1, c_2, \ldots c_n]^T$

$\underline{f}$: An nx1 Input force/torque vector = $[f_1, f_2, \ldots, f_n]^T$

Equation (2.2.1) can also be written as (see Appendix A)

$$f_i = \sum_{j=i}^{n} \Big[\sum_{k=1}^{j} \Big(Tr\Big(\frac{\partial W_j}{dq_i} J_j \frac{\partial W_j^T}{dq_k}\Big)\Big) \ddot{q}_k + \sum_{k=1}^{j} \sum_{\ell=1}^{j} \Big(Tr\Big(\frac{\partial W_j}{dq_i} J_j \frac{\partial^2 W_j^T}{\partial q_k \partial q_\ell}\Big) \dot{q}_k \dot{q}_\ell\Big) - m_j g^T \frac{\partial W_j}{\partial q_i} \, {}^j r_j \Big]$$

$$i = 1, \ldots, n \qquad (2.2.2)$$

where

J_j = Pseudo inertia tensor of link j

W_j = homogeneous (4x4) co-ordinate transformation matrix from j^{th} co-ordinate frame to the base co-ordinate frame.

$\underline{q}_j$ = as defined earlier

$\underline{\dot{q}}_j$ = as defined earlier

$\underline{\ddot{q}}_j$ = as defined earlier.

Equation (2.2.2) can be written more compactly as [3], [9]

$$f_i = \sum_{j=i}^{n} [Tr(\frac{\partial W_j}{\partial q_i} J_j \ddot{W}_j^T) - m_j g^T (\frac{\partial W_j}{\partial q_i} {}^j r_j)] \qquad i = 1,....,n \qquad (2.2.3)$$

where

$$\ddot{W}_j = \sum_{k=1}^{n} \frac{\partial W_j}{\partial q_k} \ddot{q}_k + \sum_{k=1}^{j} \sum_{\ell=1}^{j} \frac{\partial^2 W_j}{\partial q_k \, \partial_k} \dot{q}_k \dot{q}_\ell \qquad (2.2.4)$$

This formulation, greatly reduces the time required for computation. For this formulation several backward and forward recursive relations have been derived in [9]. It has also been shown in [9] that if we use 3x3 rotation matrices, instead of 4x4 rotation translation matrices, a further saving (of the order of 50%), on the operations required to be performed can be achieved.

The discrete dynamic models can also be obtained directly if it is required [7].

Forces/Torques obtained from (2.2.3) are then applied to the manipulator joints to move the end effector from initial position/orientation to the final position/orientation. However, if these are the only forces/torques acting on the links, it becomes the open loop control and it can not be predicted as to how closely the joints follow the desired trajectory. This is specially so if the joint parameters are not accurately known or if the load variation is there. Then the desired trajectory may not be followed at all. Hence there is a need to design a controller which will make the joints and, therefore, the end effector track the desired trajectory in the presence of uncertainties like load,variation in link parameters, inaccuracies because of finite computer precision, mechanical effects such as friction, vibrations etc. This controller (to be discussed in Chapter 3) has been designed on the basis of a linearized dynamic robot model and implemented as feedback. This essentially generates corrective forces/torques to those produced by the open loop control discussed before.

2.3 LINEARIZATION OF DYNAMIC EQUATION OF ROBOT

The most common technique of linearization of a nonlinear equation is using a transformation or a non-linear feedback. Several authors have reported methods of linearization of nonlinear dynamics about a point or

a set of points e.g. [8], [12]. But these techniques suffer from the drawback of inexact cancellation of non-linearity by transformation or feedback which may result in poor closed loop stability and performance.

Neuman and Murray [6] have proposed a method to linearize symbolically the Lagrangian dynamic robot model about a nominal trajectory. However the algorithms based on it cannot be run on computers (such as DEC-10) which do not have the capability of symbolic manipulation.

Balafoutis et al. [3] have presented a technique for recursive evaluation of linearized dynamic model about a nominal trajectory using sensitivity matrices. A sensitivity matrix contains sensitivity functions which, physically, characterize the perturbations of the joint co-ordinates and velocities from their nominal values in response to variations in kinematic and dynamic parameters of the links [6]. (Kinematic parameters of a link i with prismatic joint are angle between links i-1 and i, the twist angle of link i; for a revolute i^{th} joint these parameters are distance between links i-1 and i, length of link i and twist angle of link i. Dynamic link parameters appear in the pseudo-inertial matrices).

Lagrange-Euler equation of motion of a robot is given as (2.2.1) and has been rewritten as equation (2.2.3)

Let us now define

$$R_i = J_i \ddot{W}_i^T + A_{i+1} R_{i+1} \tag{2.3.1}$$

and $$\lambda_i = m_i\, {}^i r_i + A_{i+1} \lambda_{i+1} \tag{2.3.2}$$

Then equation (2.2.3) can be written as

$$f_i = Tr(\frac{\partial W_i}{\partial q_i} R_i) - g^T \frac{\partial W_i}{\partial q_i} \lambda_i \tag{2.3.3}$$

$$= \xi_i(\underline{q}, \underline{\dot{q}}, \underline{\ddot{q}}) + \Upsilon_i(\underline{q}) \tag{2.3.4}$$

where

$$\xi_i(\underline{q}, \underline{\dot{q}}, \underline{\ddot{q}}) = Tr(\frac{\partial W_i}{\partial q_i} R_i) \tag{2.3.5}$$

and $$\Upsilon(\underline{q}) = -g^T \frac{\partial W_i}{\partial q_i} \lambda_i \tag{2.3.6}$$

Equation (2.3.4) can be expanded about a nominal trajectory ($\underline{q}^o$, $\underline{\dot{q}}^o$, $\underline{\ddot{q}}^o$) ($\underline{q}^o$, $\underline{\dot{q}}^o$, $\underline{\ddot{q}}^o$ denote the nominal solution of equation(2.2.1) with nominal input force/torque $\underline{f}_o$).

The first order approximation of equation (2.3.4) around the nominal trajectory can be written as [3]

$$\underline{\delta f} = D^o \underline{\delta \ddot{q}} + V^o \underline{\delta \dot{q}} + P^o \underline{\delta q} \tag{2.3.7}$$

where $\underline{\delta f}$, $\underline{\delta q}$, $\underline{\delta \dot{q}}$ and $\underline{\delta \ddot{q}} \in R^n$ are sufficiently small deviations from the nominal values $\underline{f}^o$, $\underline{q}^o$, $\underline{\dot{q}}^o$ and $\underline{\ddot{q}}^o \in R^n$, D^o is the inertia force-acceleration sensitivity matrix evaluated about the nominal trajectory, V^o is centrifugal

and coriolis force velocity sensitivity matrix evaluated about the nominal trajectory and P^o is the force-position sensitivity matrix also evaluated about the nominal trajectory.

Let us now define the state vectors

$$\underline{x}_1 = \underline{\delta q} = [\delta q_1, \delta q_2, \ldots\ldots, \delta q_n]^T$$

$$\underline{x}_2 = \underline{\delta \dot{q}} = \dot{\underline{x}}_1 = [\delta \dot{q}_1, \delta \dot{q}_2, \ldots\ldots, \delta \dot{q}_n]^T$$

Then

$$\dot{\underline{x}}_1 = \underline{x}_2$$

and

$$\dot{\underline{x}}_2 = D^{o^{-1}}[\underline{\delta f} - V^o \underline{\delta \dot{q}} - P^o \underline{\delta q}]$$

$$= D^{o^{-1}}[\underline{\delta f} - V^o \underline{x}_2 - P^o \underline{x}_1]$$

$$= -D^{o^{-1}} P^o \underline{x}_1 - D^{o^{-1}} V^o \underline{x}_2 + D^{o^{-1}} \underline{\delta f}$$

or

$$\begin{bmatrix} \dot{\underline{x}}_1 \\ \dot{\underline{x}}_2 \end{bmatrix} = \left[\begin{array}{c|c} 0 & I \\ \hline -D^{o^{-1}} P^o & -D^{o^{-1}} V^o \end{array}\right] \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ D^{o^{-1}} \end{bmatrix} \underline{\delta f} \qquad (2.3.8)$$

or

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \qquad (2.3.9)$$

Comparison of (2.3.8) and (2.3.9) yields

$$\underline{x}^T = [\underline{x}_1^T \mid \underline{x}_2^T]^T = [\underline{\delta q}^T \mid \underline{\delta \dot{q}}^T]^T$$

$$= [\delta q_1, \ldots, \delta q_n, \delta \dot{q}_1, \ldots, \delta \dot{q}_n]^T$$

$$A = \left[\begin{array}{c|c} O & I \\ \hline -D^{o^{-1}} \; P^{o} & -D^{o^{-1}} \; V^{o} \end{array}\right]$$

and $$B = \left[\begin{array}{c} O \\ \hline -D^{o^{-1}} \end{array}\right] \; ; \; I = (nxn) \text{ Identity matrix}$$

A is, therefore, 2nx2n state matrix and B is 2nxn input matrix.

Clearly equation (2.3.8) describes a time-invariant linear model of robot about the nominal trajectory. Matrices A and B always exist because D^o is an (nxn) symmetric positive definite matrix [6]. Elements of matrices D^o, P^o and V^o can be obtained by the following equation [3]

$$d^o_{ij} = \frac{\partial \xi_i}{\partial \ddot{q}_j}\bigg|(\underline{q}^o, \underline{\dot{q}}^o, \underline{\ddot{q}}^o)$$

$$= Tr(\frac{\partial W_m}{\partial q_i} \; M_{m,j})\bigg|(\underline{q}^o, \underline{\dot{q}}^o, \underline{\ddot{q}}^o)$$

$$m = \max(i,j) \; ; \; i,j = 1,2,\ldots,n \tag{2.3.10}$$

$$M_{m,j} = J_m(\frac{\partial W^T_m}{\partial q_j}) + A_{m+1} \; M_{m+1,j}$$

$$V^o_{ij} = \frac{\partial \xi_i}{\partial \dot{q}_j}\bigg|(\underline{q}^o, \underline{q}^o, \underline{q}^o)$$

$$= 2Tr(\frac{\partial W_m}{\partial q_i} \; K_{m,j})\bigg|(\underline{q}^o, \underline{\dot{q}}^o, \underline{\ddot{q}}^o)$$

$$m = \max(i,j) ; \; i,j = 1,2,\ldots,n \tag{2.3.11}$$

$$K_{m,j} = J_m(\frac{\partial \dot{W}^T_m}{\partial q_j}) + A_{m+1} \; K_{m+1,j}$$

$$P^o_{ij} = \hat{P}^o_{ij} + \tilde{P}^o_{ij}$$

$$\hat{P}^o_{ij} = \frac{\partial \xi_i}{\partial q_j}\Big|(\underline{q}^o, \dot{\underline{q}}^o, \ddot{\underline{q}}^o)$$

$$= Tr(\frac{\partial^2 W_m}{\partial q_j \; \partial q_i} R_m + \frac{\partial W_m}{\partial q_i} \; N_{m,j})\Big|(\underline{q}^o, \dot{\underline{q}}^o, \ddot{\underline{q}}^o)$$

$$m = \max(i.j) \; ; \; i,j = 1,2,\ldots.,n$$

$$N_{m,j} = J_m \frac{\partial^2 \ddot{W}^T_m}{\partial q_j} + A_{m+1,j} \; N_{m+1,j} \qquad (2.3.12)$$

and $$\tilde{P}^o_{ij} = -g^T \frac{\partial^2 W_m}{\partial q_j \; \partial q_i} \lambda_m$$

and R_i and λ_i have been defined in equations (2.3.1) and (2.3.2).

Efficient algorithm for computing various matrices has been given in [3]. An advantage of using this algorithm is that the nominal torque (inverse dynamics) can also be calculated simultaneously.

2.4 ALGORITHM

An algorithm is given below for calculating nominal torque/force and a linearized model.

1) Read kinematic and dynamic parameters of all the links and the trajectory points for which nominal torque and linearized models should be calculated.
2) Calculate matrices relating different co-ordinate frames and their derivatives wrt time and joint variables. (An efficient algorithm for this is given in [3]).

3) Calculate nominal torque for the first trajectory point.

4) Calculate a linearized model of robot manipulator (A and B matrices) at first trajectory point.

5) If their are any more trajectory points, repeat Step 2 through 4, otherwise exit.

EXAMPLE

For the purpose of designing a controller, matrices A and B have been calculated for first three joints of PUMA-560 robot. As can be expected these matrices are dependent on load. Also a nominal torque has been calculated for a set of values of joint angles, angular velocities and accelerations. Matrices A and B for two different sets of joint angles, angular velocities and angular accelerations are presented in the Appendix C.

2.5 CONCLUSION

Problems of calculating inverse dynamics and linearization of the nonlinear dynamic equation of a robot manipulator have been considered in this chapter. It has been observed that an algorithm that does not require the cancellation of nonlinearity by a nonlinear transformation is preferable. If possible, an algorithm that produces the nominal torque alongwith the linearized model should be used to reduce computations. An algorithm has been presented and

applied for first three joints of PUMA-560 robot for obtaining the nominal torque and linearized model. It has been observed during execution of computer program that computation of linearized model and nominal control requires less than 0.5 sec for one trajectory point.

CHAPTER 3

ROBUST LQ DECENTRALIZED REGULATOR DESIGN

3.1 INTRODUCTION

The control problem of a robot arm manipulator can broadly be devided in two steps.

1) Obtaining the dynamic model of the manipulator, and
2) designing a controller.

Dynamic model of the robot has been obtained in Chapter 2.

Current industrial approaches to robot arm control system design treat each joint of the robot arm as a simple joint servomechanism [10]. This method of controlling the joints results in reduced response speed and damping and limited precision. The arm also moves with unnecessary vibrations. This is because the arm dynamics is not modelled adequately.

Various techniques have been reported e.g. [4], [10], [11], [14]-[16], for obtaining the improved speed of response and accuracy of tracking. Some of these techniques deal with resolved motion control [4], [10] and nonlinear feedback [10]. Schemes have also been presented for task oriented control of manipulators [14]. Controllers have also been designed using L_2 and L_∞ stability approaches [16] and stable factorization approach [15] and using the theory of variable structure systems [11].

The present work reports a robust LQ decentralized controller design for the control of robot arm joints. The performance of the controller is almost insensitive to the changes in the load on the end effector. The controller is easy to implement because each of the joints can be considered to be a single sub-system which is independent of other sub-systems (joints) for the purpose of implementing the controller.

The chapter has been organized in the following manner. Section 3.2 presents the methodology and the implementation details of the controller. The controller as designed in this section was used to evaluate the response of a PUMA-560 robot. The results of this evaluation are presented in Section3.3. Section 3.4 concludes the chapter with some important observations.

3.2 CONTROLLER DESIGN

The controller design consists of two parts :

1) Designing a nominal control, and
2) Designing a feedback law.

Nominal control torque can be calculated for a particular trajectory point using equation (2.3.3). Use of this equation needs the evaluation of R_i matrices and λ_i vectors. R_i matrices can be obtained using equation (2.3.1) and λ_i vectors can be obtained using equation (2.3.2). Both these equations are evaluated

in backward recursion, i.e. we start from $i = n$ (number of joints) and proceed towards $i = 1$. The important observation here is that we should assign A_{n+1} a null matrix in equations (2.3.1) and (2.3.2). Then R_i and λ_i can be calculated for i varying from n to 1. Other matrices required for calculating the nominal torque are also required for obtaining a linear error equation of robot manipulator around that point. Hence it is advantageous to calculate nominal torque while obtaining linear error model.

The nominal torque propels the joints to the neighbourhood of the desired trajectory points. A decentralized feedback control law is designed for the whole trajectory such that it retains the optimality and robustness for all the trajectory points under various conditions of loading and thereby ensures excellent tracking accuracies. The control system is then implemented as a MIMO system with decentralized feedback. The decentralized feedback control law has been designed using the method developed in [13]. A brief review of this method is given below :

Let the equation of the robot arm motion be given as

$$S_p : \dot{\underline{x}} = A_p\underline{x} + B\underline{u} \qquad (3.2.1)$$

where x is an n state vector, u is an m-input vector, A_p is an nxn state matrix and B is an nxm input matrix.

A_p and B are both constant matrices.

This system can be decomposed as below treating each joint as a sub-system :

$$S_{ip} : \quad \dot{\underline{x}}_i = A_{ii}\underline{x}_i + B_i\underline{u}_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} A_{ij}\underline{x}_j \qquad (3.2.2)$$

where $i = 1,2,\ldots\ldots, r$ (No. of joints)

$\underline{x}_i$ = n_i-state vector

$\underline{u}_i$ = m_i input vector

A_{ii} = $n_i x n_i$ state matrix

B_i = $n_i x m_i$ input matrix

A_{ij} = $n_i x n_j$ interconnection matrix.

The term $\sum_{\substack{j=1 \\ j\neq i}}^{n} A_{ij}x_j$ represents the coupling of other joints with i^{th} joint. Neglecting this term,

$$\dot{\underline{x}}_i = A_{ii}\underline{x}_i + B_i\underline{u}_i \qquad (3.2.3)$$

Writing a compact equation for the complete system neglecting interconnections;

$$\dot{\underline{x}} = A_D\underline{x} + B_D\underline{u}$$

where $\underline{x}^T = [\underline{x}_1^T, \underline{x}_2^T, \ldots\ldots, \underline{x}_r^T]^T$

and $\underline{u}^T = [\underline{u}_1^T, \underline{u}_2^T, \ldots\ldots, \underline{u}_r^T]^T$

A_D and B_D are block diagonal matrices defined as

$$A_D = \text{diag}\,(A_1,\ldots.,A_r)$$

$$B_D = \text{diag}\,(B_1,\ldots.,B_r)$$

where,

$$A_i = \begin{bmatrix} A_{i1} & A_{i2},\ldots\ldots, & A_{in_i} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ A_{n_{i1}} & A_{n_{i2}},\ldots\ldots, & A_{n_i n_i} \end{bmatrix} ; \quad B_i = \begin{bmatrix} B_{i1} & \ldots\ldots & B_{im_i} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ B_{in_i} & \ldots\ldots & B_{n_i m_i} \end{bmatrix}$$

Let a matrix A_C contain all the interconnection terms. Then assuming that the pair $[A_D,B_D]$ is completely controllable and the pair $[A_D, D]$ is completely observable, the control law obtained in [13] is

$$\underline{u}^* = -L_\alpha \underline{x}$$

where L_α is given by

$$L_\alpha = \Sigma^{-1} B_D^T P_\alpha \tag{3.2.4}$$

P_α is the nxn real unique p.d. symmetric matrix and is the solution of the algebric Riccati equation (ARE)

$$A_\alpha^T P_\alpha + P_\alpha A_\alpha - P_\alpha B \Sigma^{-1} B^T P_\alpha + Q = 0 \tag{3.2.5}$$

and $$A_\alpha = A_D + \alpha I, \quad \alpha > 0 \tag{3.2.6}$$

This control law minimizes the quadratic cost function

$$J_\alpha = \int_0^\infty e^{2\alpha t}(\underline{x}^T Q \underline{x} + \underline{u}^T \Sigma \underline{u})\, dt \qquad (3.2.7)$$

where

$$Q = \text{diag}\,(Q_1, \ldots\ldots, Q_r)$$

$$\Sigma = \text{diag}\,(\Sigma_1, \ldots\ldots, \Sigma_r)$$

Q_i = $n_i x n_i$ p.s.d. state weighing matrix

Σ_i = $m_i x m_i$ p.d. input weighing matrix

and α in (3.2.6) is so chosen that the matrix F_α is atleast n.s.d. where

$$F_\alpha = A_c^T P_\alpha + P_\alpha A_c - Q - 2\alpha P_\alpha \qquad (3.2.8)$$

The scalar α is defined as the degree of stability [1] which physically means that the closed loop poles of the system are placed to the left of $S = -\alpha$ in the S-plane.

The initial gauss of α can be obtained from the following equation

$$\alpha \geq \frac{1}{2} \lambda_{max} [\bar{Q}\, P_o^{-1}] \qquad (3.2.9)$$

where

$\lambda_{max}[\cdot]$ is maximum eigen-value of matrix $[\cdot]$, and

$\bar{Q} = A_c^T P_o + P_o A_c - Q$ where, P_o is the solution of ARE (3.2.5) with $\alpha = 0$ in (3.2.6).

The value of α which makes F_α in (3.2.8) atleast n.s.d. when used in J_α in (3.2.7) results in a feedback gain matrix L_α which when used in full model with interconnections guarantees optimality of the resulting closed-loop system [13]. The optimality in turn guarantees excellent robustness properties for the closed-loop system.

A program has been written using this method. The program needs the maximum load expected and the trajectory points around which the controller should be obtained. It calculates the controller for different load conditions (e.g. no load condition, maximum load condition and half load condition) at various trajectory points and evaluates which controller retains the robustness and outputs that controller. An algorithm for obtaining the controller is given below :

1) Obtain the linearized models around the desired trajectory for no load.
2) Obtain matrices A_c, A_D and B_D.
3) Choose $\alpha = 0$ and obtain P_o as a solution of ARE (equation 3.2.5).
4) Obtain a conservative estimate of α using (3.2.9).
5) Substitute α in (3.2.6) and obtain P_α as a solution of (3.2.5).
6) For this α and P_α, check if F_α (equation 3.2.8) is n.s.d. or not.

7) If F_α is n.d., decrease the value of α and repeat steps (5) and (6). If F_α is p.d., increase the value of α and repeat steps (5) and (6).

8) Repeat step (7) till a minimum value of α is obtained such that F_α is n.s.d.

9) Repeat steps (1) to (8) for different load conditions.

10) For each controller obtained above, check which controller retains optimality for all loads and for all trajectory points. (This can be done by checking whether F_α in equation 3.2.8 remains n.s.d. or not).

11) Output the controller that retains optimality for all loads and for all trajectory points.

3.3 EXAMPLE

A nominal control and a feedback control gain matrix were designed for a PUMA-560 robot for the evaluation of control strategy followed in the present work. The evaluation has been carried out for first three joints only.

Figs. (3.3.1a) to (3.3.1c) present nominal trajectories for the robot arm joints which they are required to follow. A nominal control was designed for no load condition of robot arm for these trajectories. If unloaded arm is driven by this nominal control, the tracking performance will obviously be satisfactory. To find the degradation in the performance with changes in load, performance was observed by driving the arm with this nominal control when loaded with a cubic

load of 6 Kg weight and 0.1 m dimension.

The performance of the robot was observed for two seconds in the open loop. As seen from the Fig. (3.3.2), the arm moves with large vibrations and does not settle to a final position even after two seconds. Also the maximum error in tracking for the joints is as high as 0.48 rad, 0.41 rad and 1.00 rad for first, second and third joints respectively.

Feedback torque is now applied to the joints and the performance of the controller observed for three different loads. Fig. (3.3.3) presents the performance of the controller under no load condition. It is observed that the maximum error in tracking for the first joint is around 0.022 rad and the joint finally settles to the final value 0.0043 rad in 0.93 sec. Similarly maximum error for joint two is of the order of 0.021 rad and it settles to 0.0062 rad in 0.93 sec. Joint 3, however, has larger error which is of the order of 0.04 rad and it settles to 0.0059 rad in 0.93 sec.

The arm was now loaded with a cubic load of 3 Kg weight and dimension 0.1 m. Errors in trajectory tracking have been plotted in Fig. (3.3.4) as a function of time. A very marginal degradation in performance is observed with maximum tracking errors of 0.026 rad, 0.025 rad and 0.065 rad for joints one, two and three respectively. But the steady state error has converged to the same values in the same time interval as obtained in no load case (Fig. 3.3.3).

The load was then changed to cubic load of 6 Kg weight and 0.1 m dimension. Fig. (3.3.5) presents the error in tracking as a function of time in this load condition. Again a marginal degradation of performance is observed in the sense of maximum error in tracking (0.031 rad, 0.033 rad, and 0.13 rad respectively for joints one, two and three). But the steady state errors converges to the same value as obtained in no load case in the same duration of time.

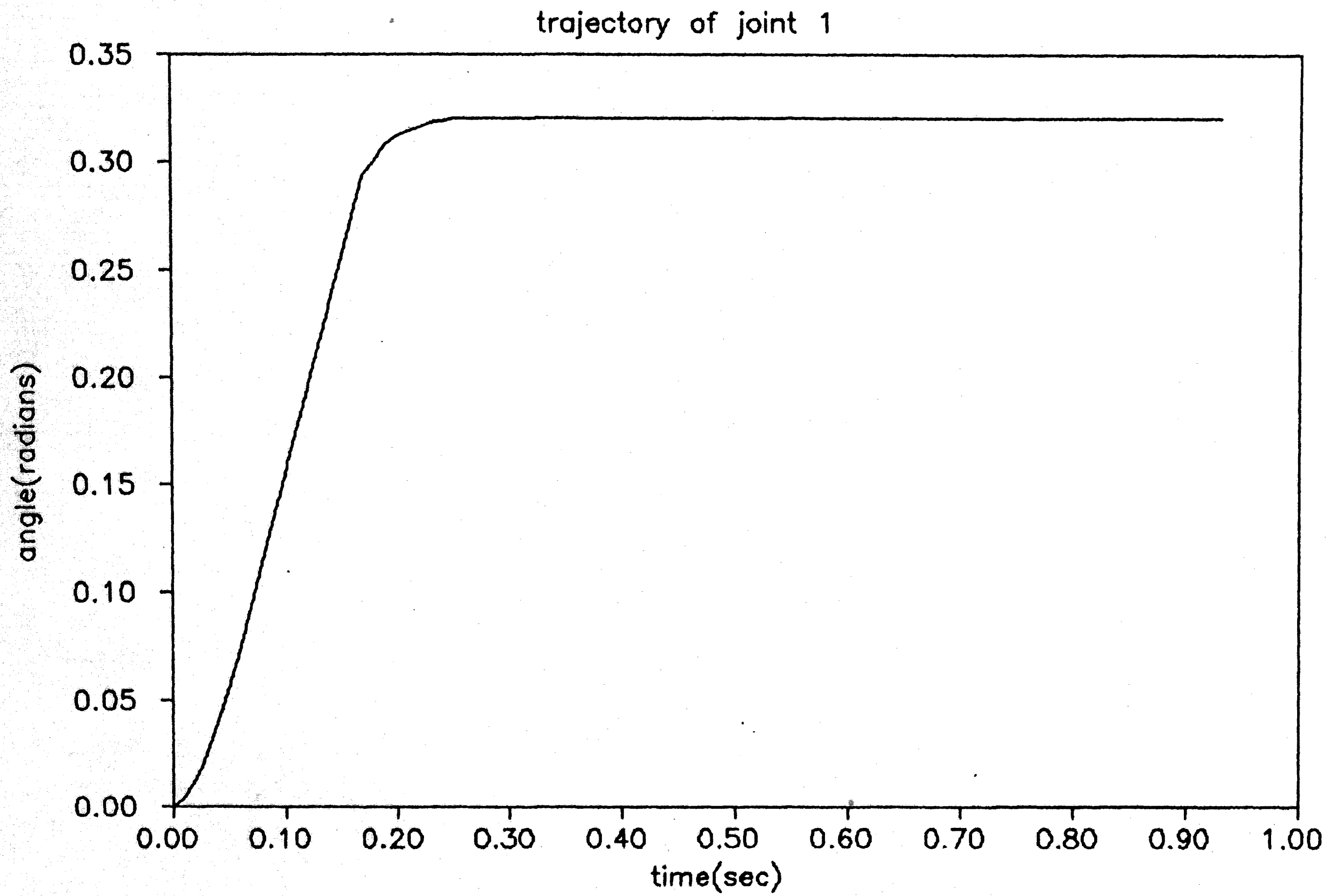
3.4 DISCUSSION AND CONCLUSION

It was observed in Section 3.2 that the controller designed as described there retains optimality property under various load conditions. The optimality property in turn guarantees excellent robustness properties like infinite gain margin and 60° phase margin for the closed-loop system. Figs. (3.3.3) - (3.3.5) clearly show that controller retains the satisfactory performance when the load is changed from no load to 6 Kg load of 0.1 m dimension including all intermediate load conditions. Only a marginal degradation in the maximum tracking errors is observed.

Section 3.2 also indicates the importance of parameter α. It was observed during the design of the controller that the value of α required to make F_{α} matrix (equation 3.2.8) n.s.d. is largest for no load condition at that point on the trajectory where maximum velocities and accelerations are encountered. This point represents the worst case condition. Therefore, the controller designed for such a condition will

preserve optimality property for various load conditions.

It may be worth comparing the tracking accuracies obtained using this controller with some other methods of controlling the manipulators. Slotine [17] has reported a comparision of tracking errors of proportional derivative (p.d.),computed torque and adaptive controllers. Comparision of the maximum tracking errors under no load condition suggests that the performance of adaptive controller, computed torque controller and the one designed in this thesis are comparable whereas p.d. controllers definitely seems to be inferior with larger tracking errors. When the arm in [17] is loaded, tracking errors increase. This increase in errors is much higher than that experienced in the case of LQ controller designed here. Steady state errors also increase significantly in [17] whereas no appreciable increase in steady state errors has been observed in the present case. So in this respect the performance of the robust decentralized LQ controller designed in this thesis seems to be superior. Added to this superiority of performance is the ease in implementing this controller. Whereas adaptive controllers are very difficult to implement, the decentralized, constant LQ controller is very easy to implement. Since there is only one controller for the whole trajectory, there is no need to calculate the controller on line. Once the trajectory planning has been done, controller can be designed off-line and implemented very cost-effectively unlike in the case of adaptive controllers.



trajectory of joint 1
angle(radians)
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
time(sec)

Fig. 3.3.1a

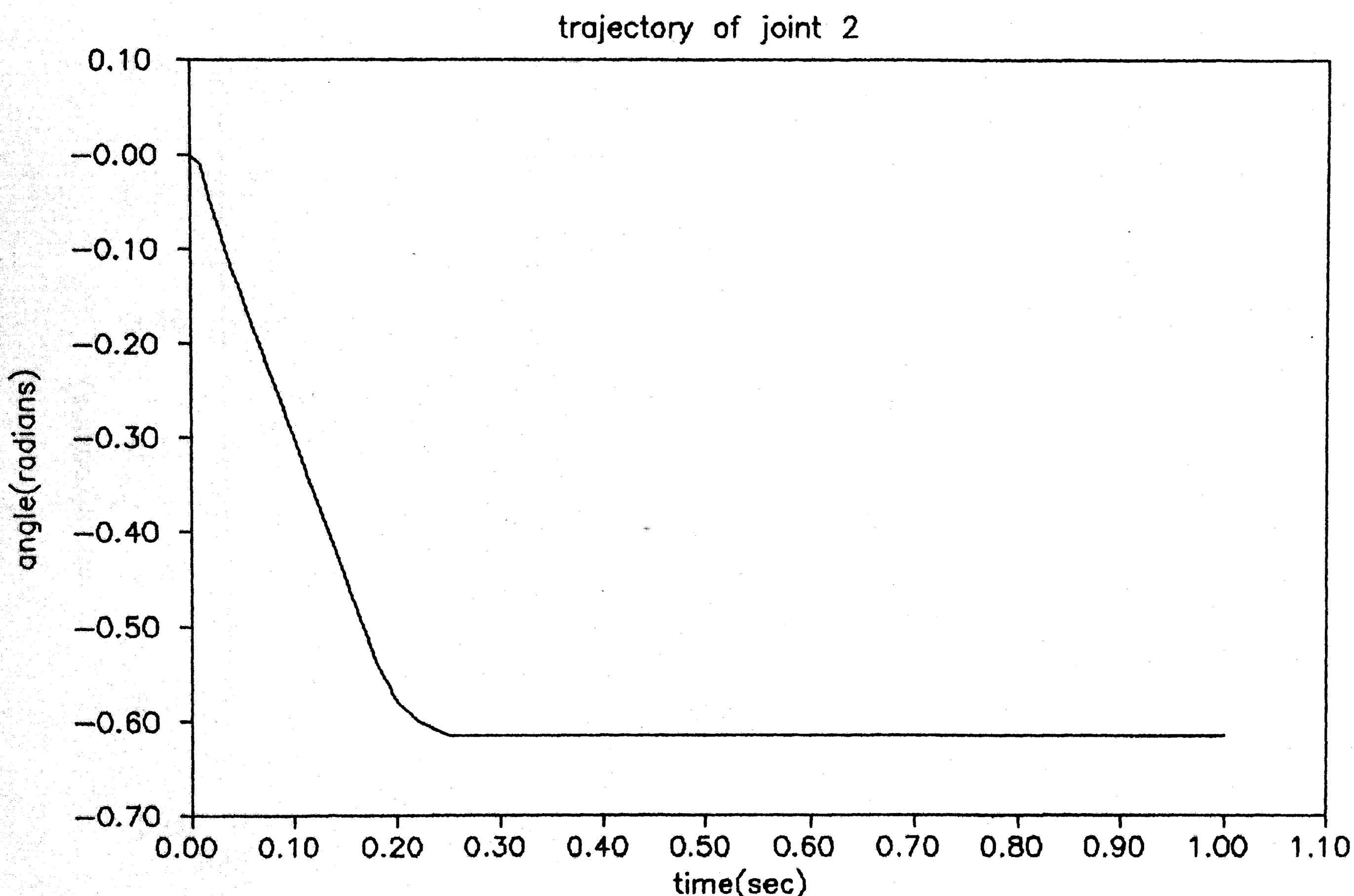

trajectory of joint 2
0.10
-0.00
-0.10
-0.20
-0.30
-0.40
-0.50
-0.60
-0.70
angle(radians)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
1.10
time(sec)

Fig. 3.3.1b

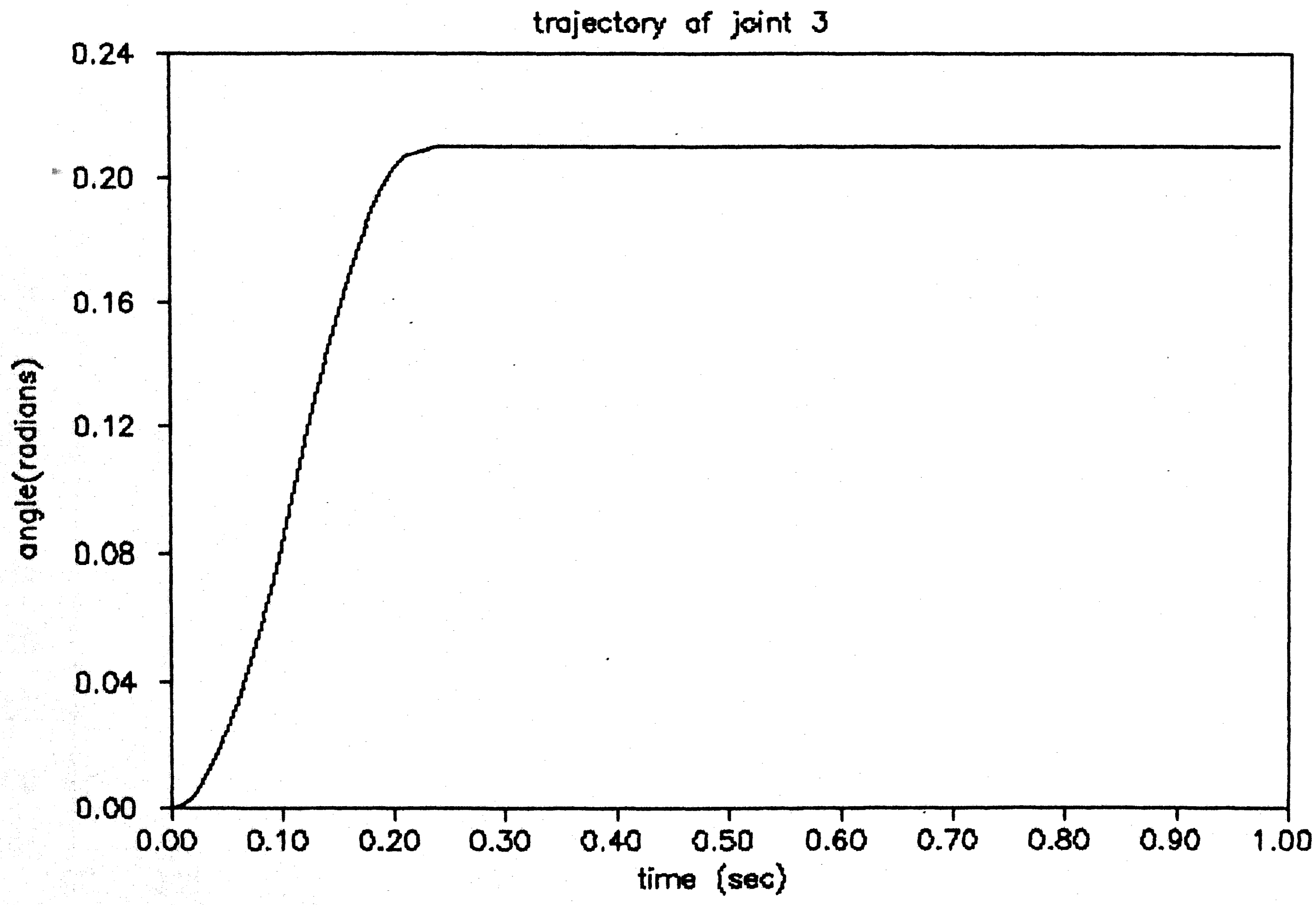

trajectory of joint 3
angle(radians)
0.24
0.20
0.16
0.12
0.08
0.04
0.00
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
time (sec)

Fig. 3.3.1c

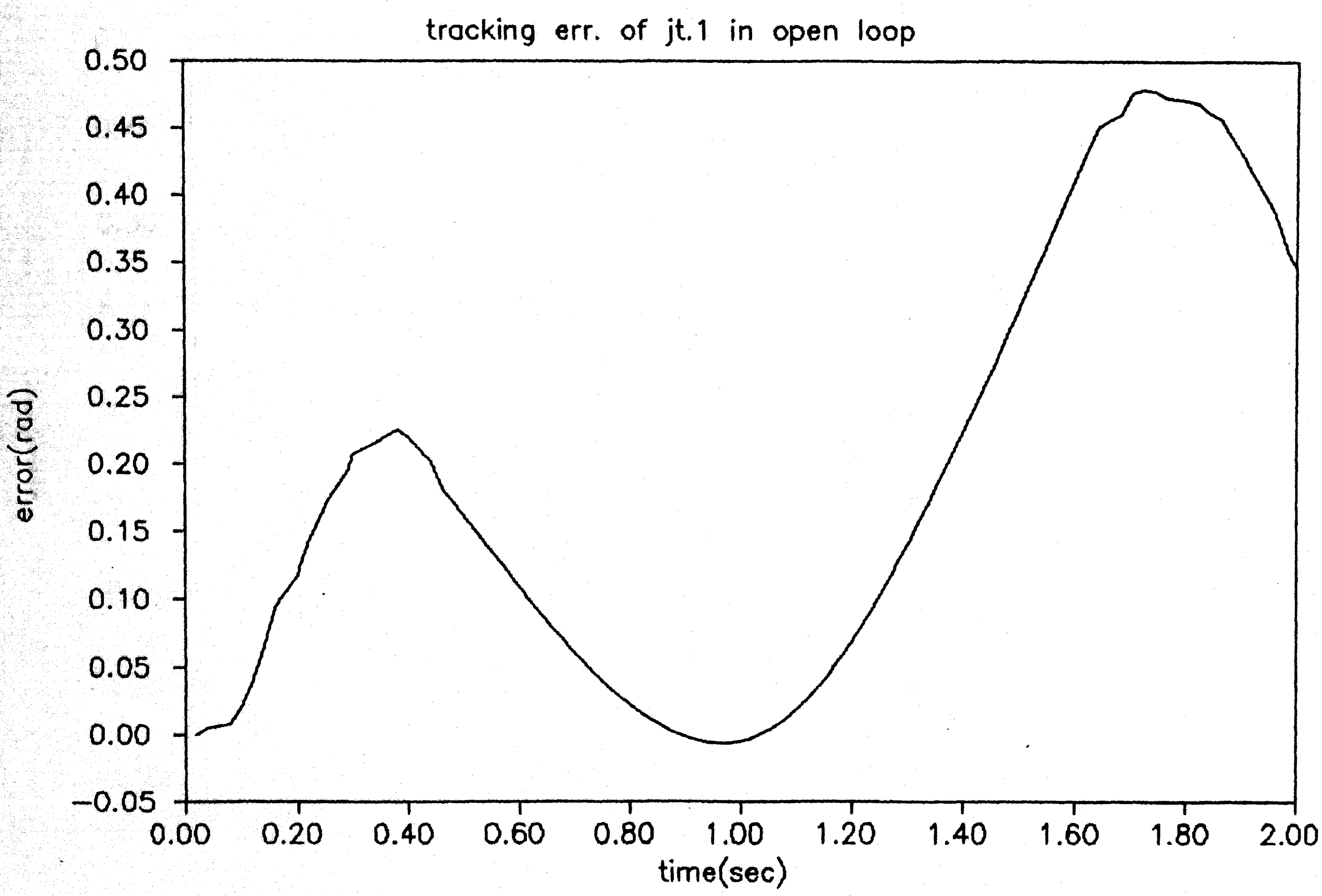

tracking err. of jt.1 in open loop
error(rad)
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80
2.00
time(sec)

Fig. 3.3.2a

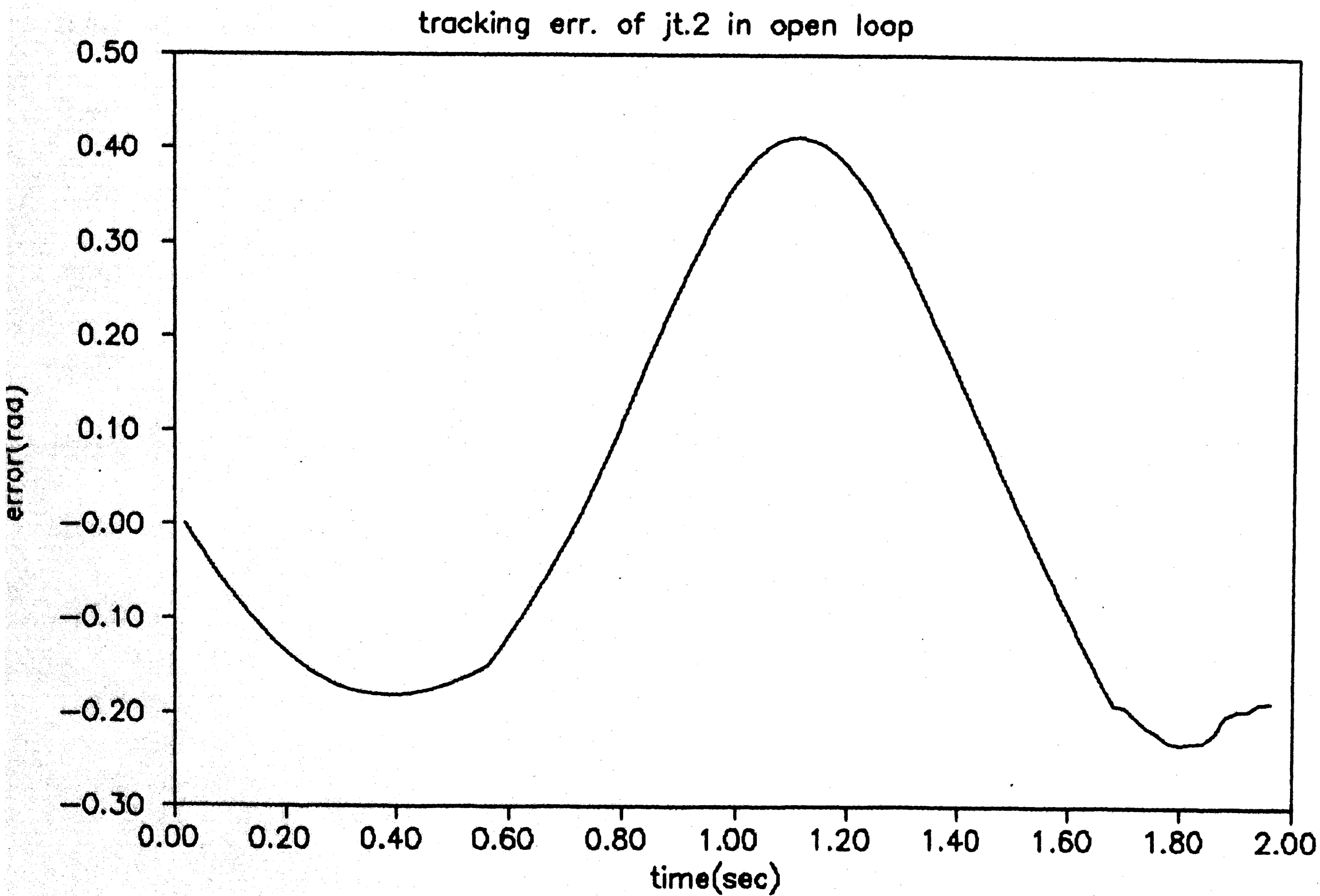

tracking err. of jt.2 in open loop
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
error(rad)
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80
2.00
time(sec)

Fig. 3.3.2b

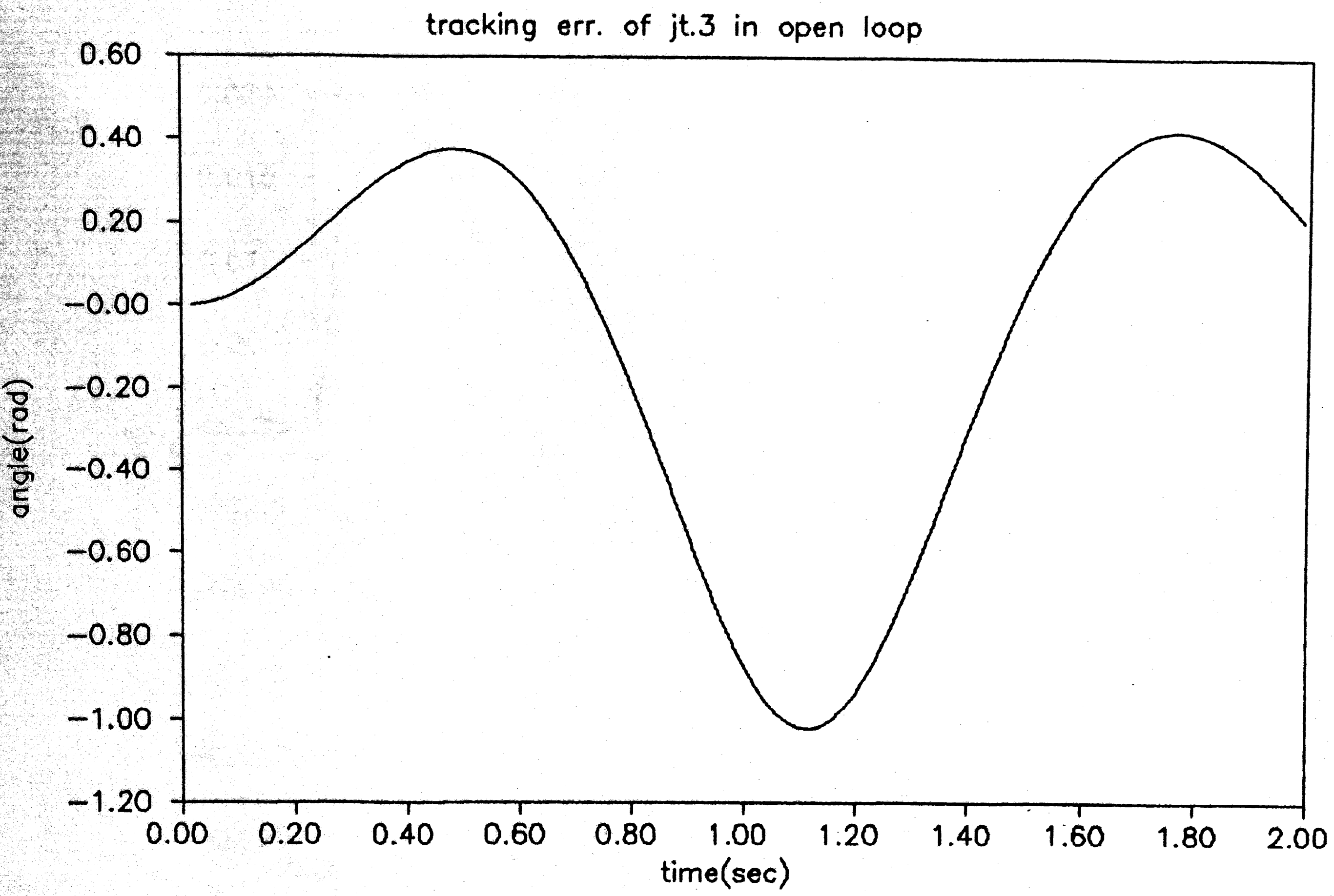

tracking err. of jt.3 in open loop
0.60
0.40
0.20
-0.00
-0.20
-0.40
-0.60
-0.80
-1.00
-1.20
angle(rad)
0.00
0.20
0.40
0.60
0.80
1.00
1.20
1.40
1.60
1.80
2.00
time(sec)

Fig. 3.3.2c

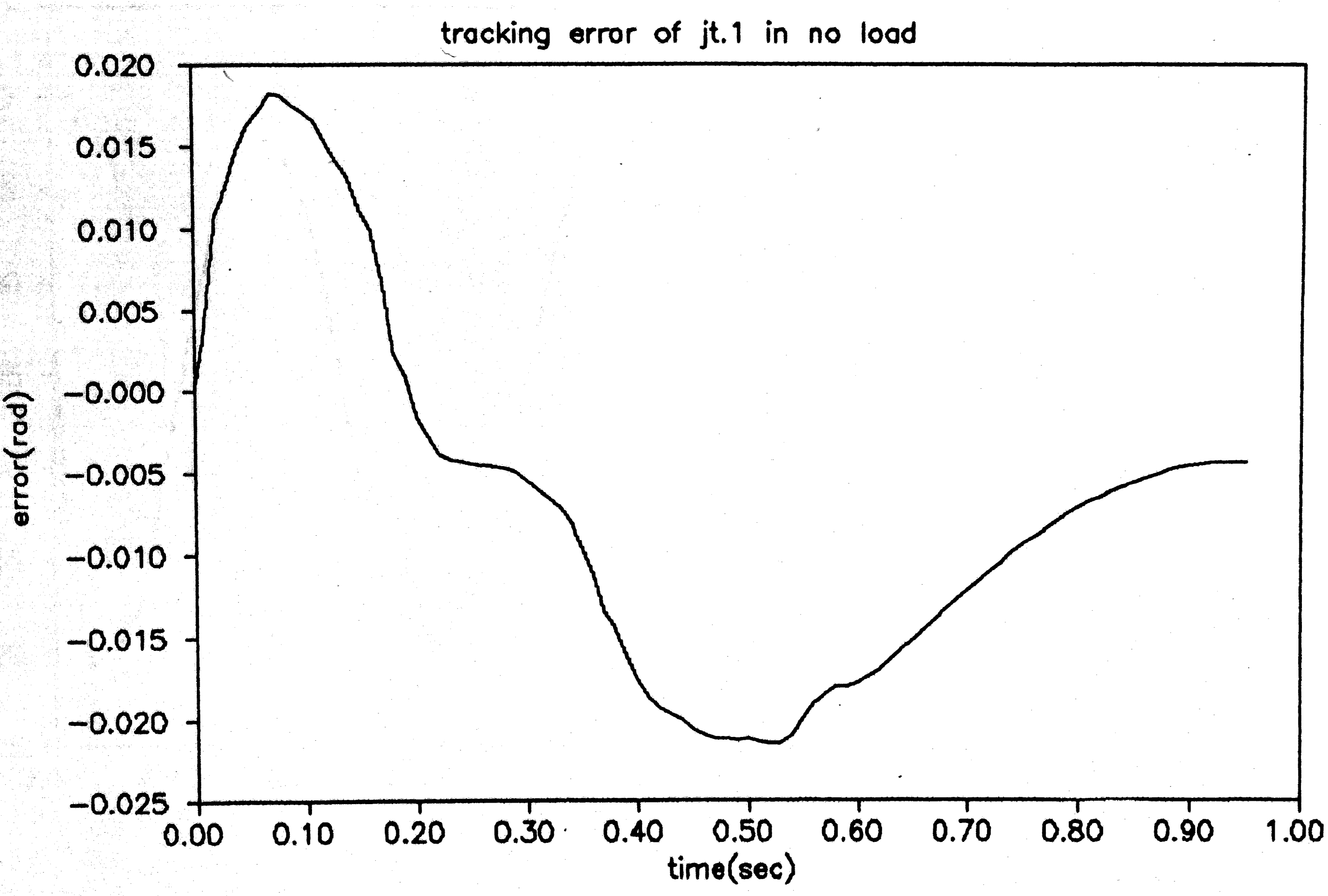

tracking error of jt.1 in no load
0.020
0.015
0.010
0.005
−0.000
−0.005
−0.010
−0.015
−0.020
−0.025
error(rad)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
time(sec)

Fig. 3.3.3a

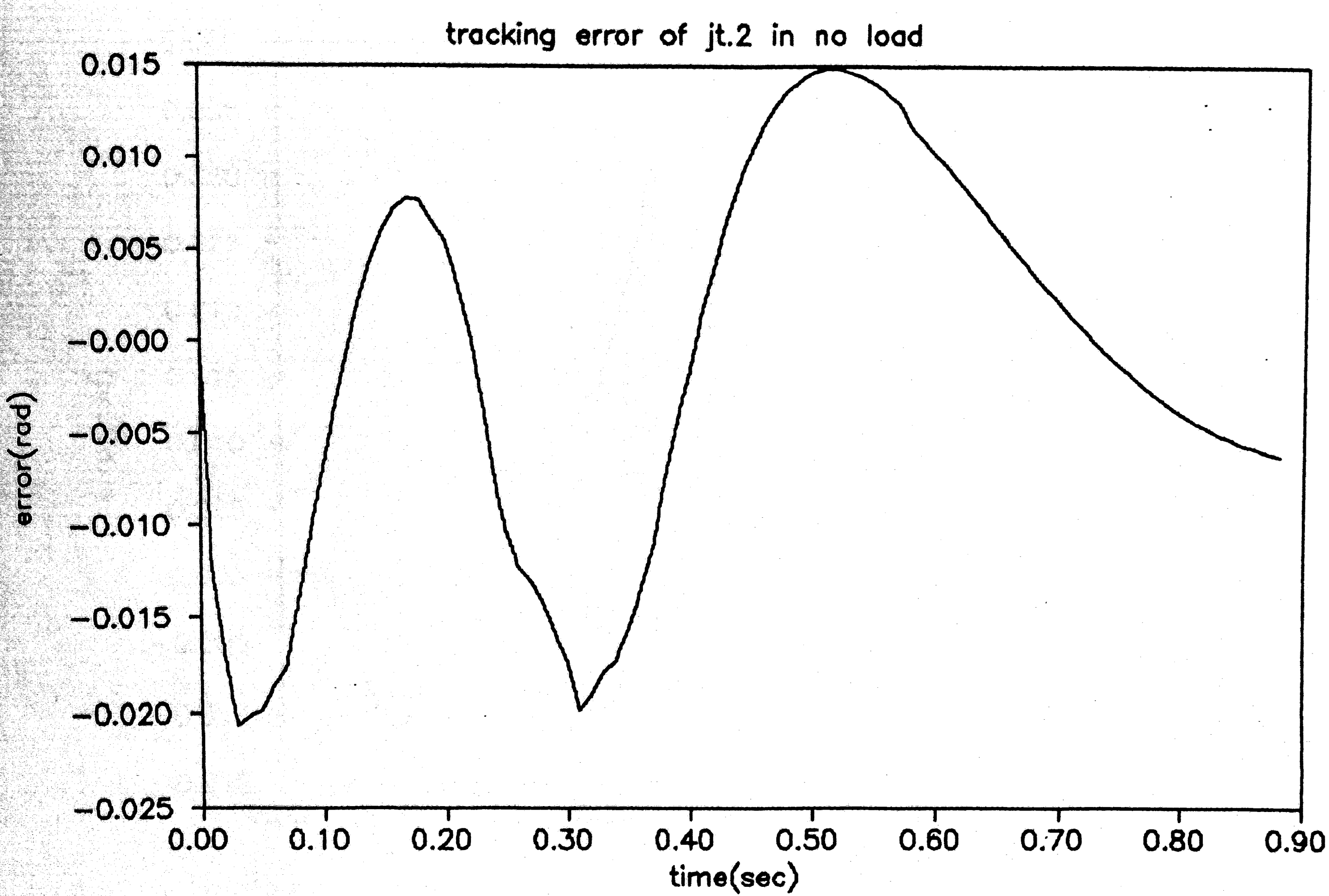
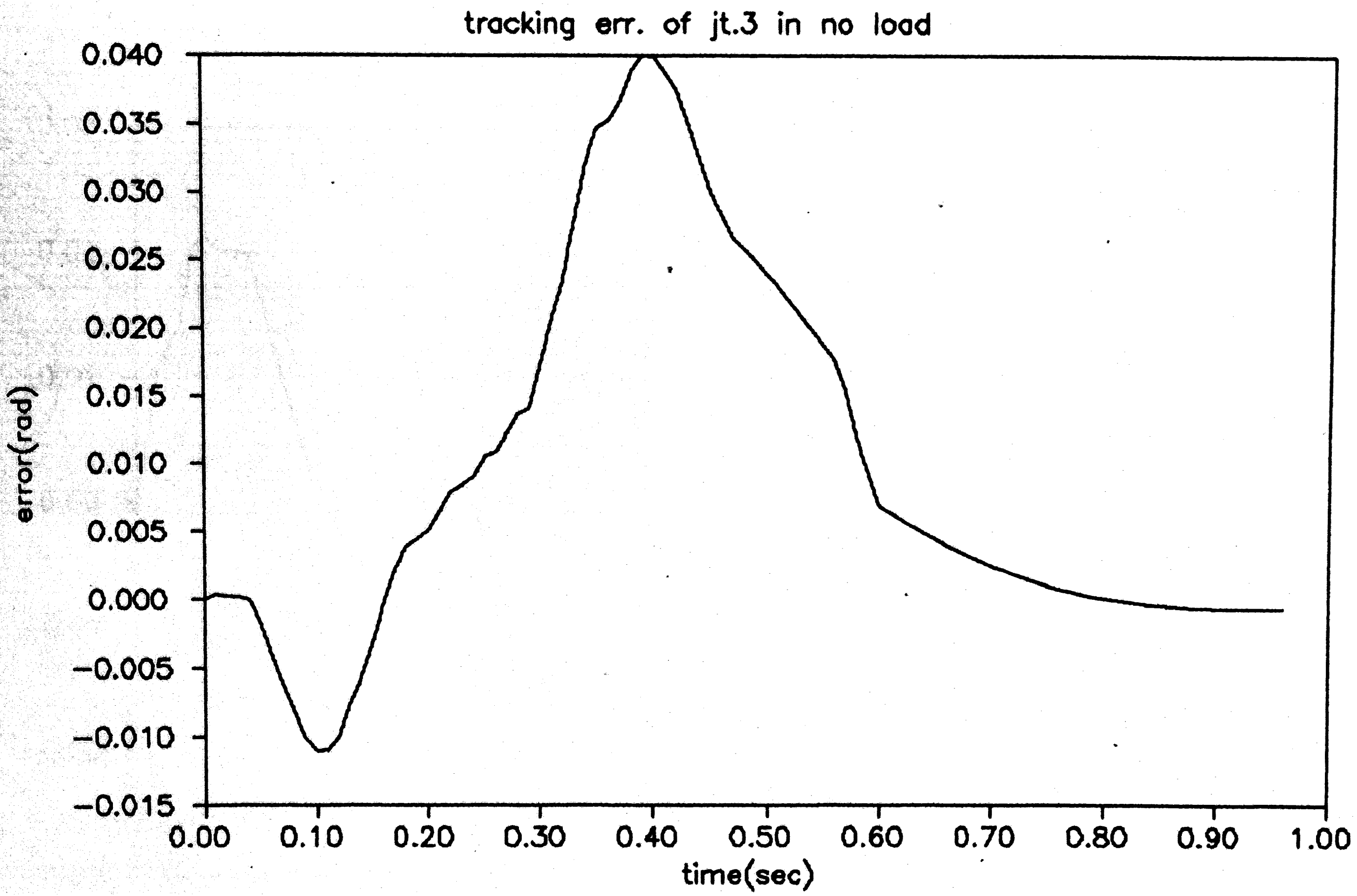

tracking error of jt.2 in no load
0.015
0.010
0.005
-0.000
-0.005
-0.010
-0.015
-0.020
-0.025
error(rad)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
time(sec)

Fig. 3.3.3b



tracking err. of jt.3 in no load
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
error(rad)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
time(sec)

Fig. 3.3.3c

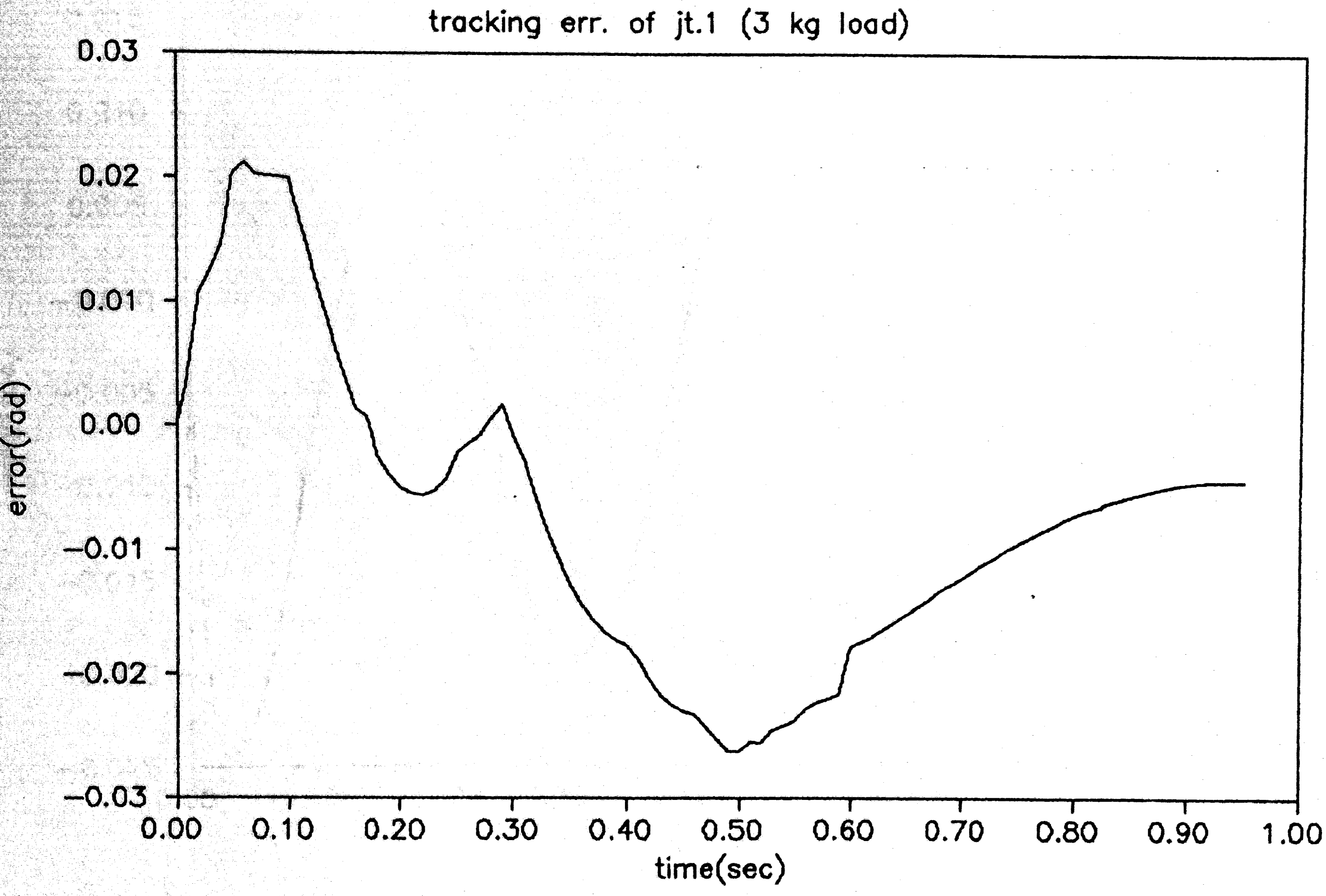

tracking err. of jt.1 (3 kg load)
error(rad)
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
time(sec)

Fig. 3.3.4a

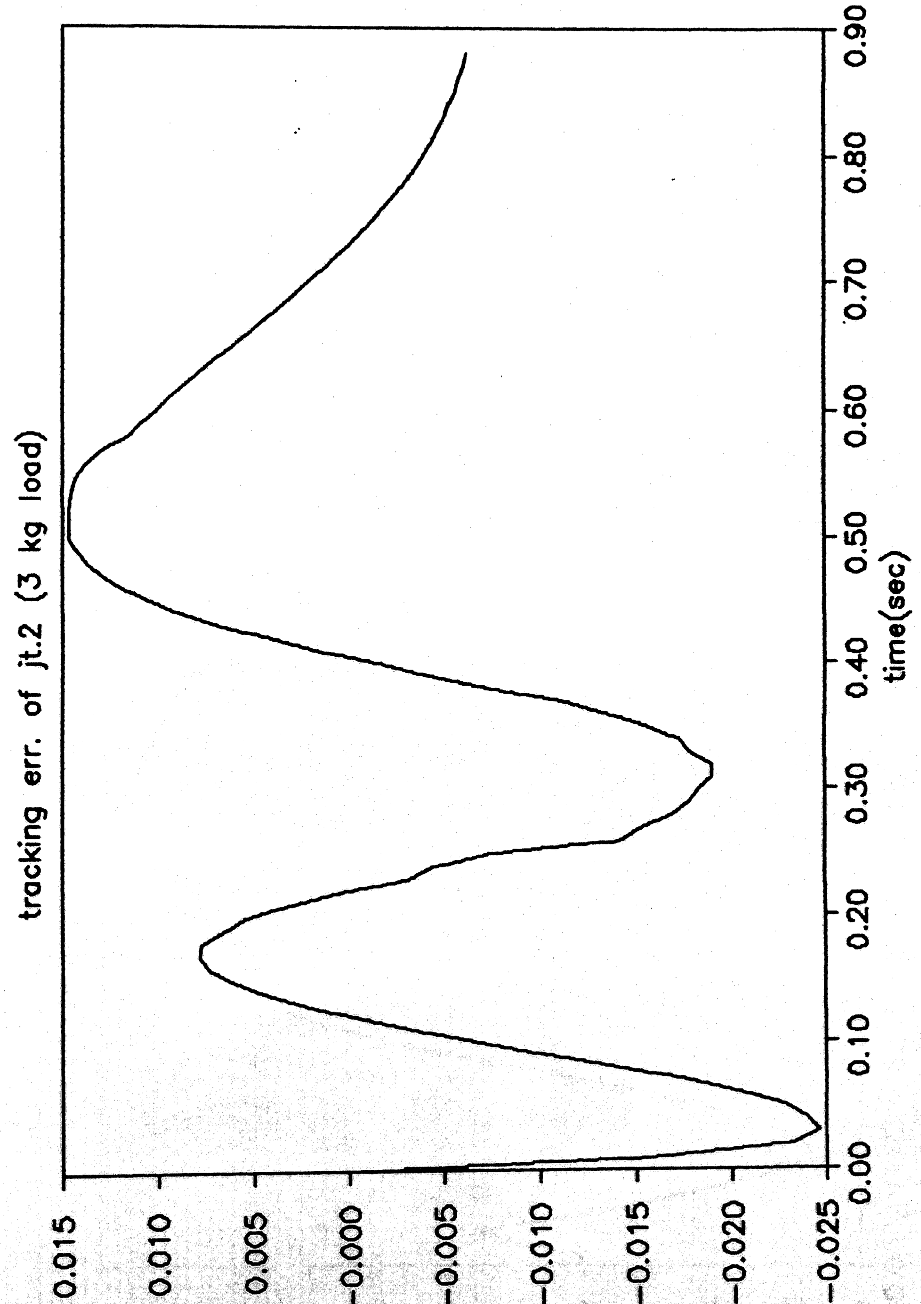

tracking err. of jt.2 (3 kg load)
error(rad)
0.015
0.010
0.005
-0.000
-0.005
-0.010
-0.015
-0.020
-0.025
time(sec)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90

Fig. 3.3.4b

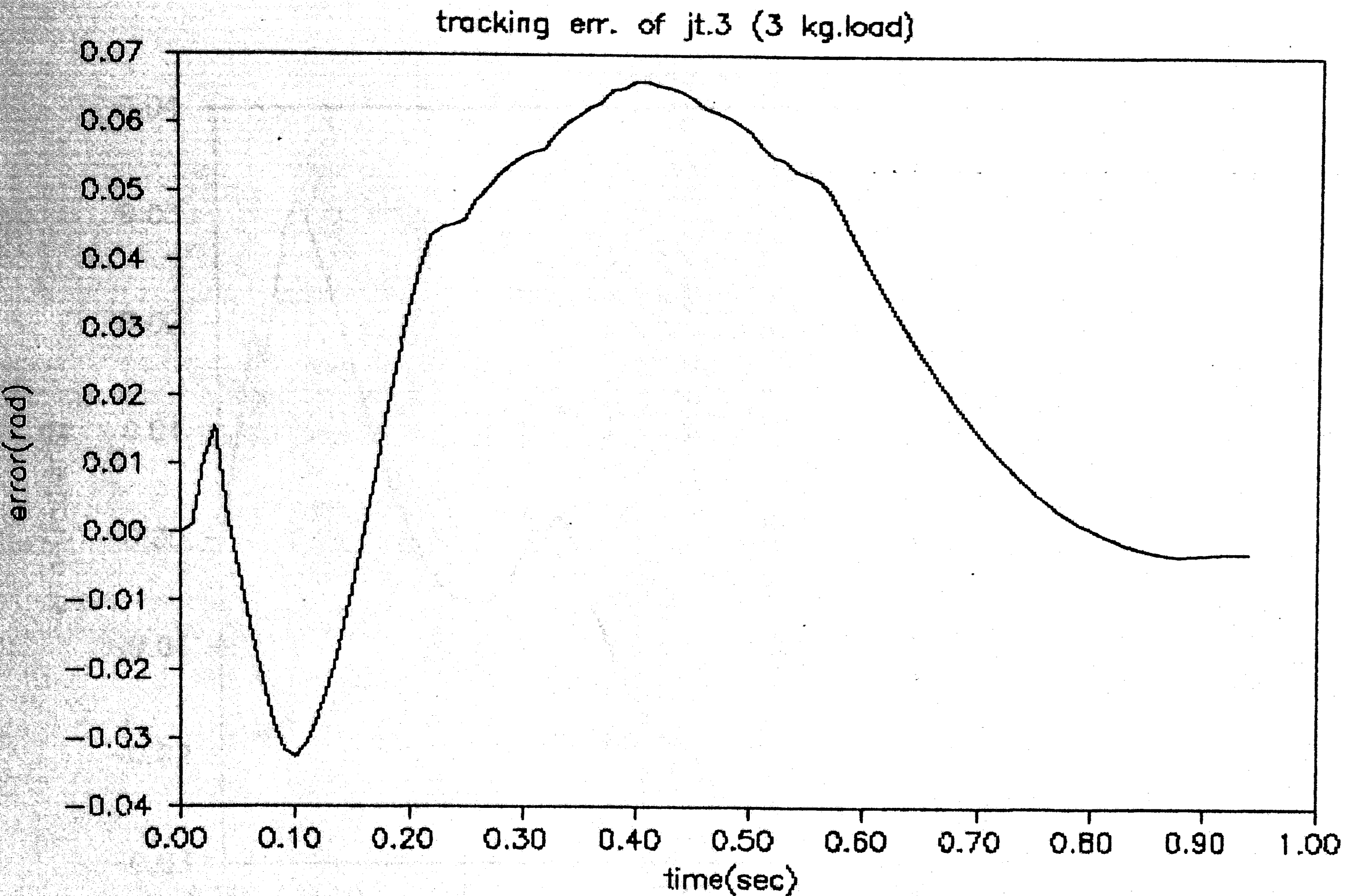

tracking err. of jt.3 (3 kg.load)
error(rad)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
time(sec)

Fig. 3.3.4c

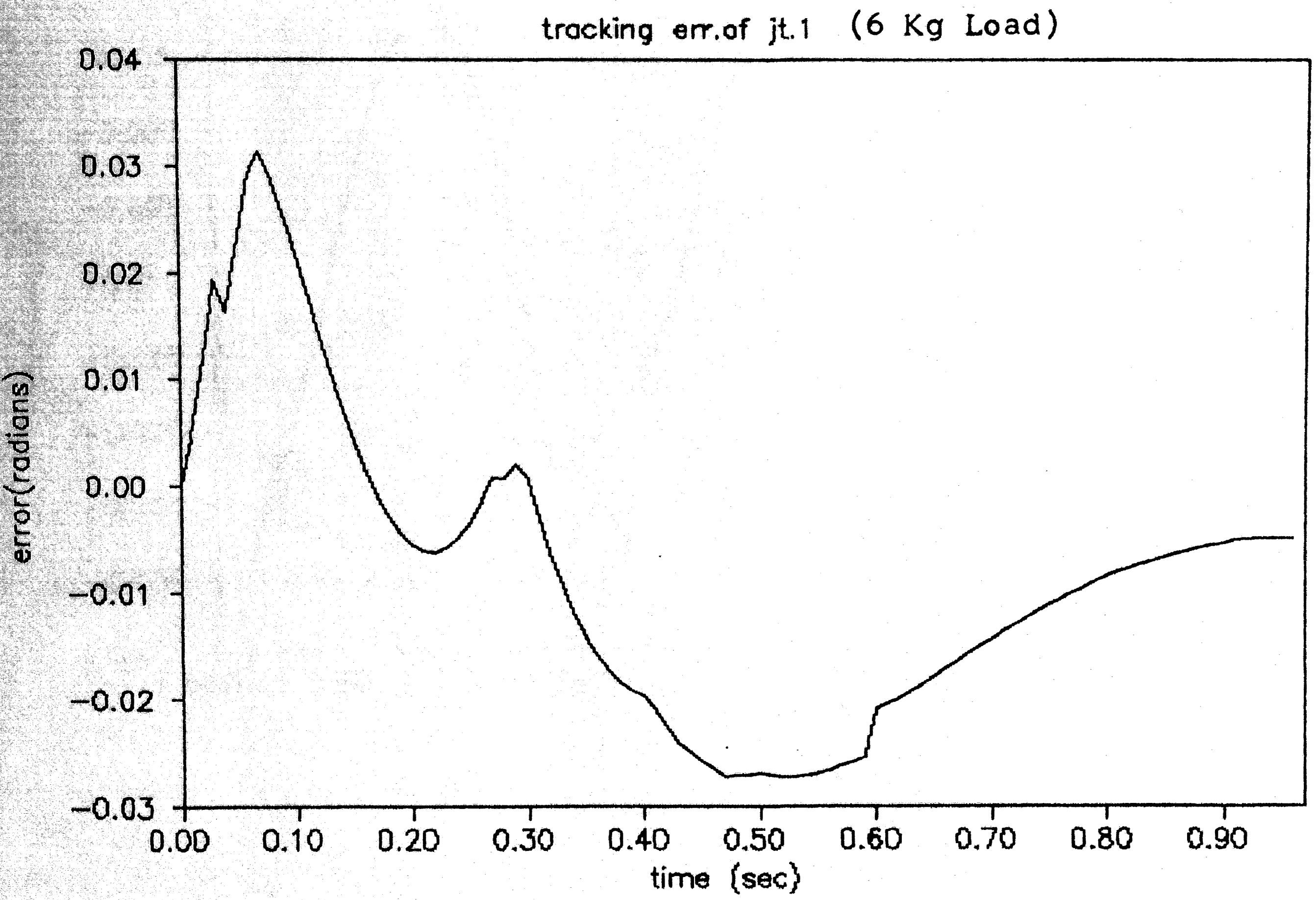

tracking err.of jt.1 (6 Kg Load)
error(radians)
0.04
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
time (sec)

Fig. 3.3.5a

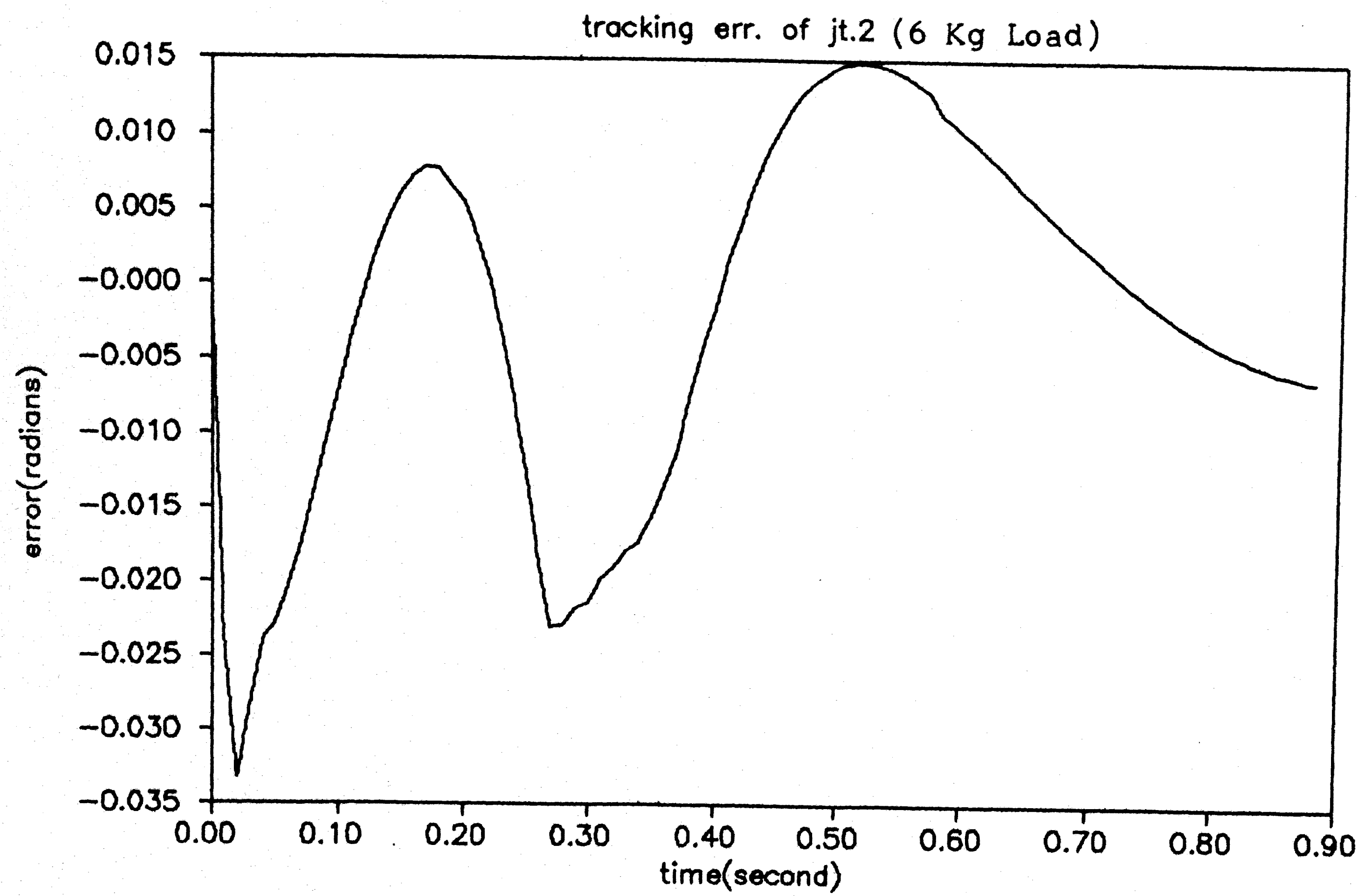

tracking err. of jt.2 (6 Kg Load)
error(radians)
0.015
0.010
0.005
-0.000
-0.005
-0.010
-0.015
-0.020
-0.025
-0.030
-0.035
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
time(second)

Fig. 3.3.5b

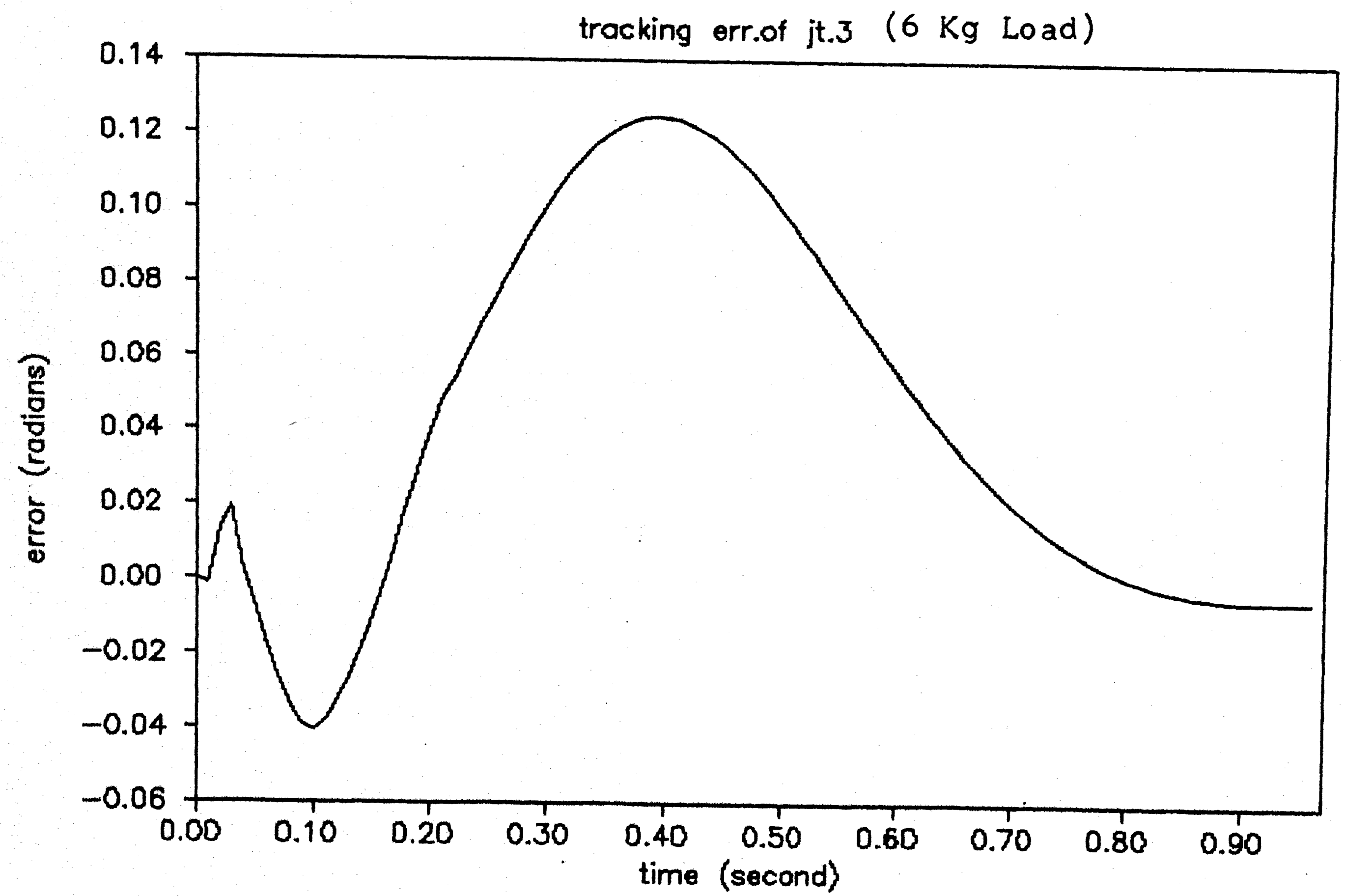

tracking err.of jt.3 (6 Kg Load)
error (radians)
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
time (second)

Fig. 3.3.5c

CHAPTER 4

CONCLUSION

Main results of the thesis have been summarized in this chapter.

In Chapter two, the problem of the evaluation of robot arm dynamics was considered. Linearized error models about a trajectory have also been obtained. Finally an algorithm has been given for finding the linearized models around the trajectory. It was also pointed out that the nominal control can also be obtained while evaluating the linearized models.

In Chapter three, the problem of designing a robust controller was considered. A method to design the decentralized LQ regulator was presented and a controller for PUMA-560 robot was designed. The performance of the controller was evaluated for various load conditions. It was observed that a controller designed for no load and for a trajectory point where highest velocities and accelerations are encountered retains optimality of the closed-loop system. This, in term, guarantees the robustness of performance. This fact has been verified in this chapter when it is seen that under various load conditions, there is no significant change in the performance.

It is worth noting here, that the controller as designed in Chapter three is a decentralized controller which makes the physical implementation of the controller very easy while resulting in a satisfactory performance. Although while finding out the controller all three joints were considered together, procedure is equally suited for designing the controller for individual joints if desired.

REFERENCES

[1] Brian D.O. Anderson and John B. Moore, ''Linear Optimal Control,'' Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[2] B.R. Patnaik, ''Computed Torque Adaptive Control of Robot Manipulators'', M.Tech. Thesis, I.I.T. Kanpur, 1987.

[3] C.A. Balafoutis, P. Mishra and R.V. Patel, ''Recursive Evaluation of Linearized Dynamic Robot Models'', IEEE Journal of Robotics and Automation, Vol. RA-2, No. 3, pp. 146-154, 1986.

[4] Chi-Haur Wu and Richard P. Paul, ''Resolved Motion Force Control of Robot Manipulator'', IEEE Trans. Systems, Man and Cybernatics, Vol. SMC-12, No. 3, pp. 266-275, 1982.

[5] C.L. Li, ''A New Method of Dynamics for Robot Manipulators'', IEEE Trans. Systems, Man and Cybernatics, Vol. 18, No. 1, pp. 105-114, 1988.

[6] C.P. Neuman and J.J. Murray, ''Linearization and Sensitivity Functions of Dynamic Robot Models'', IEEE Trans. Systems, Man and Cybernatics, Col. SMC-14, No. 6, pp. 805-818, 1984.

[7] C.P. Neuman and V.D. Tourassis, ''Discrete Dynamic Robot Models'', IEEE Trans. Systems, Man and Cybernatics, Vol. SMC-15, No. 2, pp. 193-204, 1985.

[8] C. Reboulet and C. Champetier, ''A New Method for Linearizing Non-Linear Systems : The Pseudo Linearization'', Int. J. Control, Vol. 40, No. 4, pp. 631-638, 1984.

[9] J.M. Hollerbach, ''A Recursive Lagrangian Formulation of Manipulator Dynamics and A Comparative Study of Dynamics Formulation Complexity'', IEEE Trans. Systems, Man and Cybernatics, Vol. SMC-10, No. 11, pp. 730-736, 1980.

[10] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, Robotics, Control, Sensing, Vision and Intelligence, McGraw Hill Co., Singapore.

[11] K.S. Yeung and Y.P. Chen, ''A New Controller Design for Manipulators Using the Theory of Variable Structures'', IEEE Trans. Automatic Control, Vol. 33, No. 2, pp. 200-206, Feb. 1988.

[12] L.R. Hunt, R. Su and G. Meyer, ''Global Transformations of Non-Linear Systems'', IEEE Trans. Automatic Control, Vol. AC-28, No. 1, pp. 24-31, 1983.

[13] Manju Dixit, ''Design of Optimality Preserving Decentralized Controllers for LTI Interconnected Systems'', M.Tech. Thesis, I.I.T. Kanpur, 1988.

[14] Monkazu Takegaki and Suguru Arimoto, ''A New Feedback Method for Dynamic Control of Manipulators'', ASME Dynamic Systems Measurement and Control, Vol. 102, pp. 109-125, June 1981.

[15] M.W. Spong and M. Vidyasagar, ''Robust Linear Compensator Design for Non-linear Robotic Control'', IEEE J. Robotics and Automation, Vol. RA-3, No.4, pp.345-351, Aug. 1987.

[16] N. Becker and W.M. Grimm, ''On L_2^- and L_∞^- Stability Approaches for Robust Control of Robot Manipulators'', IEEE Trans. Automatic Control, Vol. 33, No. 1, pp. 118-122, Jan. 1988.

[17] J.J.E. Slotine and W. Li, ''Adaptive Manipulator Control : A Ca e Study'', IEEE Trans. Automatic Control, Vol. 33, No. 11, pp. 995-1003, Nov. 1988.

APPENDIX A

L-E equation of motion of a robot arm manipulator as given in (2.2.1) is rewritten here as equation (A.1)

$$D(\underline{q})\,\ddot{\underline{q}} + \underline{h}(\underline{q}, \dot{\underline{q}}) + \underline{c}(\underline{q}) = \underline{f} \qquad (A.1)$$

This equation can also be written as [10]

$$\sum_{k=1}^{n} d_{ik}\, \ddot{q}_k + \sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm}\, \dot{q}_k\, \dot{q}_m + c_i = f_i \qquad (A.2)$$

where

$$d_{ik} = \sum_{j=\max(i,k)}^{n} \mathrm{Tr}\left(\frac{\partial W_j}{\partial q_k}\, J_j\, \frac{\partial W_j^T}{\partial q_i}\right) \qquad (A.3)$$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^{n} \mathrm{Tr}\left(\frac{\partial^2 W_j}{\partial q_k\, \partial q_m}\, J_j\, \frac{\partial W_j^T}{\partial q_i}\right) \qquad (A.4)$$

and $$c_i = \sum_{j=i}^{n} \left(-m_j\, g^T\, \frac{\partial W_j}{\partial q_i}\, {}^j r_j\right) \qquad (A.5)$$

for $i,k,m = 1,2,\ldots,n$

and g = gravity vector $[g_x, g_y, g_z, 0]^T$

${}^j r_j$ = Co-ordinate of the centre of mass of the j^{th} link expressed in its own co-ordinate frame.

Define $U_{ij} = \dfrac{\partial W_i}{\partial q_j}$ and $U_{ijk} = \dfrac{\partial^2 W_i}{\partial q_j\, \partial q_k}$

Then

$$\sum_{k=1}^{n} d_{ik}\,\ddot{q}_k = \sum_{k=1}^{n}\ \sum_{j=\max(i,k)}^{n} \mathrm{Tr}\,(U_{ik}\ J_j\ U_{ji}^T)\ddot{q}_k$$

Let us expand this term for $i = 1$ and $n = 3$

$$\sum_{k=1}^{3} d_{1k}\,\ddot{q}_k = \sum_{k=1}^{3}\ \sum_{j=k}^{3} \mathrm{Tr}(U_{jk}\ J_j\ U_{j1}^T)\ddot{q}_k$$

$$= \mathrm{Tr}(U_{11}J_1U_{11}^T)\ \ddot{q}_1 + \mathrm{Tr}(U_{21}J_2U_{21}^T)\ \ddot{q}_1$$

$$+ \mathrm{Tr}(U_{31}J_3U_{31}^T)\ \ddot{q}_1 + \mathrm{Tr}(U_{22}J_2U_{21}^T)\ \ddot{q}_2$$

$$+ \mathrm{Tr}(U_{32}J_3U_{31}^T)\ \ddot{q}_2 + \mathrm{Tr}(U_{33}J_3U_{31}^T)\ \ddot{q}_3$$

Rearranging the terms above

$$\sum_{k=1}^{3} d_{1k}\,\ddot{q}_k = \mathrm{Tr}(U_{11}J_1U_{11}^T)\ \ddot{q}_1 + \mathrm{Tr}(U_{21}J_2U_{21}^T)\ \ddot{q}_1$$

$$+ \mathrm{Tr}(U_{21}J_2U_{22}^T)\ \ddot{q}_2 + \mathrm{Tr}(U_{31}J_3U_{31}^T)\ \ddot{q}_1$$

$$+ \mathrm{Tr}(U_{31}J_3U_{32}^T)\ \ddot{q}_2 + \mathrm{Tr}(U_{31}J_3U_{33}^T)\ \ddot{q}_3$$

$$= \sum_{j=1}^{n}\ \sum_{k=1}^{j} \mathrm{Tr}(U_{jk}J_jU_{ji}^T)\ q_k$$

Also noting that trace of a matrix and its transpose are equal, above equation can be written as

$$\sum_{k=1}^{3} d_{1k}\,\ddot{q}_k = \sum_{j=1}^{n}\ \sum_{k=1}^{j} \mathrm{Tr}(U_{j1}J_jU_{jk}^T)\ \ddot{q}_k$$

Similarly for $i = 2$, it can be shown that

$$\sum_{k=1}^{3} d_{2k}\ \ddot{q}_k = \sum_{j=2}^{n} \sum_{k=1}^{j} Tr(U_{j2} J_j U_{jk}^T)\ \ddot{q}_k$$

and so on. Hence

$$\sum_{k=1}^{n} d_{ik}\ \ddot{q}_k = \sum_{j=i}^{n} \sum_{k=1}^{j} Tr(U_{ji} J_j U_{jk}^T)\ \ddot{q}_k \qquad (A.6)$$

Similarly,

$$\sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm}\ \dot{q}_k \dot{q}_m = \sum_{k=1}^{n} \sum_{m=1}^{n} \sum_{j=\max(i,k,m)}^{n} Tr(U_{jkm} J_j U_{ji}^T)$$

For i = 1 and n = 3, above equation

$$\sum_{k=1}^{3} \sum_{m=1}^{3} h_{1km}\ q_k q_m = \sum_{k=1}^{3} \sum_{m=1}^{3} \sum_{j=\max(i,k,m)}^{3} Tr(U_{jkm} J_j U_{j1}^T) \dot{q}_k \dot{q}_m$$

$$= Tr(U_{111} J_1 U_{11}^T) \dot{q}_1^2 + Tr(U_{211} J_2 U_{21}^T)\ \dot{q}_1^2$$

$$+ Tr(U_{311} J_3 U_{31}^T)\ \dot{q}_1^2 + Tr(U_{212} J_2 U_{21}^T)\ \dot{q}_1 \dot{q}_2$$

$$+ Tr(U_{312} J_3 U_{31}^T)\ \dot{q}_1 \dot{q}_2 + Tr(U_{313} J_3 U_{31}^T)\ \dot{q}_1 \dot{q}_3$$

$$+ \ldots\ldots + Tr(U_{331} J_3 U_{31}^T) \dot{q}_3 \dot{q}_1 + Tr(U_{332} J_3 U_{31}^T) \dot{q}_3 \dot{q}_2$$

$$+ Tr(U_{333} J_3 U_{31}^T)\ \dot{q}_3^2$$

Rearranging the terms above,

$$\sum_{k=1}^{3} \sum_{m=1}^{3} h_{1km}\, \dot{q}_k \dot{q}_m = Tr(U_{111} J_1 U_{11}^T)\, \dot{q}_1^2 + Tr(U_{211} J_2 U_{21}^T) \dot{q}_1^2$$

$$+ Tr(U_{212} J_2 U_{21}^T) \dot{q}_1 \dot{q}_2 + Tr(U_{221} J_2 U_{21}^T) \dot{q}_2 \dot{q}_1$$

$$+ Tr(U_{222} J_2 U_{21}^T)\, \dot{q}_2^2 + Tr(U_{311} J_3 U_{31}^T) \dot{q}_1^2$$

$$+ Tr(U_{312} J_3 U_{31}^T)\, \dot{q}_1 \dot{q}_2 + Tr(U_{313} J_3 U_{31}^T) \dot{q}_1 \dot{q}_3$$

$$+ Tr(U_{321} J_3 U_{31}^T) \dot{q}_2 \dot{q}_1 + Tr(U_{322} J_3 U_{31}^T) \dot{q}_2^2$$

$$+ Tr(U_{323} J_3 U_{31}^T) \dot{q}_2 \dot{q}_3 + Tr(U_{331} J_3 U_{31}^T) \dot{q}_3 \dot{q}_1$$

$$+ Tr(U_{332} J_3 U_{31}^T) \dot{q}_3 \dot{q}_2 + Tr(U_{333} J_3 U_{31}^T) \dot{q}_3 \dot{q}_2$$

$$= \sum_{j=1}^{n} \sum_{k=1}^{j} \sum_{\ell=1}^{j} (Tr(U_{jk\ell}\, J_j U_{ji}^T)\, \dot{q}_k \dot{q}_\ell)$$

$$= \sum_{j=1}^{n} \sum_{k=1}^{j} \sum_{\ell=1}^{j} (Tr(U_{ji} J_i U_{jk\ell}^T)\, \dot{q}_k \dot{q}_\ell)$$

Similarly for i = 2 it can be shown that

$$\sum_{k=1}^{3} \sum_{m=1}^{3} h_{2km}\, q_k q_m = \sum_{j=2}^{n} \sum_{k=1}^{j} \sum_{\ell=1}^{j} (Tr(U_{ji} J_j U_{jk}^T)\, \dot{q}_k\, \dot{q}_\ell)$$

and so on. Hence

$$\sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm}\, q_k q_m = \sum_{j=i}^{n} \sum_{k=1}^{j} \sum_{=1}^{j} (Tr(U_{ji} J_i U_{jk}^T) q_k\; q\;) \tag{A.7}$$

And from (A.5) we have

$$c_i = \sum_{j=i}^{n} (-m_j \, g^T \, U_{ji} \, {}^j r_j) \tag{A.8}$$

Substituting for $\sum_{k=1}^{n} d_{ik} \, \ddot{q}_k$, $\sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm} \, \dot{q}_k \dot{q}_m$ and c_i

from equations (A.6) and (A.7) and (A.8) respectively into eq. (A.2), we have

$$\sum_{j=i}^{n} \sum_{k=1}^{j} Tr(U_{ji} J_i U_{jk}^T) \, \ddot{q}_k + \sum_{j=i}^{n} \sum_{k=1}^{j} \sum_{\ell=1}^{j} (Tr(U_{ji} J_j U_{jk\ell}^T) \dot{q}_k \dot{q}_\ell)$$

$$+ \sum_{j=i}^{n} (-m_j \, g^T \, U_{ji} \, {}^j r_j) = f_i$$

or

$$\sum_{j=i}^{n} [\sum_{k=1}^{j} (Tr(\frac{\partial W_j}{\partial q_i} \, J_j \, \frac{\partial W_j^T}{\partial q_k}) \, \ddot{q}_k)$$

$$+ \sum_{k=1}^{j} \sum_{\ell=1}^{j} (Tr(\frac{\partial W_j}{\partial q_i} \, J_j \, \frac{\partial^2 W_j}{\partial q_k \, \partial q}) \, \dot{q}_k \dot{q}_\ell - m_j \, g^T \, \frac{\partial W_j}{\partial q_i} \, {}^j r_j]= f_i$$

APPENDIX B

Link Parameter and Mass Properties of PUMA-560 Robot

1.1 Link Parameters

Joint Number	θ_i (in degrees)	α_i (in degrees)	a_i (in meters)	d_i (in meters)
1	-160° to 60°	-90°	0	0
2	-225° to $+45^\circ$	0°	0.4318	0.1495
3	-45° to 225°	90°	0	0

1.2 Mass Properties

Inertia tensors of first three links of PUMA-560 are given below :

$\text{diag } J_1 = [0.0071, 0.0267, 0.0267] \text{ Kg m}^2$

$\text{diag } J_2 = [0.1000, 0.7300, 0.8025] \text{ Kg m}^2$

$\text{diag } J_3 = [0.0222, 0.2160, 0.2245] \text{ Kg m}^2$

1.3 Centres of Gravity of Links

First link $[0.0, 0.0, 0.073]^T$ m

Second link $[-0.4318, 0.0, 0.0]^T$ m

Third link $[0.0, 0.0, 0.1]^T$ m

1.4 Mass of Links

$m_1 = 2.27$ Kg

$m_2 = 15.97$ Kg

$m_3 = 11.36$ Kg

APPENDIX C

Trajectory Point 1 :

$q = [0.2,\ 0.2,\ 0.2]^T$ rad   Nominal torque = [5.289 -46.034 -4.629

$q = [1,\ 1,\ 1]^T$ rad/s

$q = [1,\ 1,\ 1]^T$ rad/s^2

NO-LOAD CONDITION

A-MATRIX

0.0000000E+00	1.000000	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.2618776E-09	0.5052836	-1.012501	0.4358792	-0.1191027	-0.2342148
0.0000000E+00	0.0000000E+00	0.0000000E+00	1.000000	0.0000000E+00	0.0000000E+00
0.4047156E-08	5.139830	2.366487	-0.7057863	-0.8486320	-0.6262366
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	1.000000
-0.6787522E-08	-36.09692	-33.83332	3.278110	4.050746	0.9416308

B-MATRIX

0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2438718	-0.2343241E-01	0.1351310
0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.2343241E-01	0.3621334	-0.6073372
0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1351310	-0.6073373	4.047957

3 Kg LOAD CONDITION

A-MATRIX

0.0000000E+00	1.000000	0.0000000E+00	0.0000000E+00
-0.1733694E-09	0.5216651	-0.9939919	0.1343015
0.0000000E+00	0.0000000E+00	0.0000000E+00	1.000000
0.2695879E-08	5.102533	2.328399	-0.7026666
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.4493965E-08	-35.57309	-33.29985	3.234465

0.0000000E+00	0.0000000E+00
-0.1207158	-0.2341027
0.0000000E+00	0.0000000E+00
-0.8447819	-0.6258833
0.0000000E+00	1.000000
3.996607	0.9364464

B-MATRIX

0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2435136	-0.2326925E-01	0.1330774
0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.2326925E-01	0.3618347	-0.6031698
0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1330774	-0.6031698	3.989610

6 Kg LOAD CONDITION

A-MATRIX

0.0000000E+00	1.000000	0.0000000E+00	0.0000000E+00
0.7912618E-09	0.5527586	-0.9586863	0.1312866
0.0000000E+00	0.0000000E+00	0.0000000E+00	1.000000
-0.1743076E-08	5.031367	2.255713	-0.6967125
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2446058E-07	-34.57340	-32.28176	3.151171

0.0000000E+00	0.0000000E+00
-0.1237700	-0.2338642
0.0000000E+00	0.0000000E+00
-0.8374358	-0.6252104
0.0000000E+00	1.000000
3.893287	0.9265528

B-MATRIX

0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2428057	-0.2295645E-01	0.1291578
0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.2295645E-01	0.3612648	-0.5952165
0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1291578	-0.5952165	3.878259

TRAJECTORY POINT NO. = 2

$q = [0, 0, 0]^T$ rad Nominal torque = $[0, -48.124, 0]^T$

$q = [0, 0, 0]^T$ rad/s

$q = [0, 0, 0]^T$ rad/s^2

A-MATRIX

$$\begin{bmatrix} 0.0000000E+00 & 1.000000 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & -1.414231 & -1.414231 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 1.000000 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 1.120994 & 1.120994 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 1.000000 \\ 0.0000000E+00 & -33.99983 & -33.99983 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \end{bmatrix}$$

B-MATRIX

$$\begin{bmatrix} 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.2594140 & -0.4331147E-02 & 0.1313640 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ -0.4331147E-02 & 0.3458301 & -0.4465231 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.1313640 & -0.4465231 & 3.500546 \end{bmatrix}$$

3 Kg LOAD CONDITION

A-MATRIX

$$\begin{bmatrix} 0.0000000E+00 & 1.000000 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & -1.391847 & -1.391847 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 1.000000 \\ 0.0000000E+00 & 1.104652 & 1.104652 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & -33.50417 & -33.50417 & 0.0000000E+00 \end{bmatrix}$$

0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	1.000000
0.0000000E+00	0.0000000E+00

B-MATRIX

0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2589986	-0.4262594E-02	0.1292848
0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.4262594E-02	0.3457801	-0.4450051
0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1292848	-0.4450051	3.454506

6 Kg LOAD CONDITION

A-MATRIX

0.0000000E+00	1.000000	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.0000000E+00	-1.348994	-1.348994	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00	0.0000000E+00	1.000000	0.0000000E+00	0.0000000E+00
0.0000000E+00	1.073357	1.073357	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	1.000000
0.0000000E+00	-32.55499	-32.55499	0.0000000E+00	0.0000000E+00	0.0000000E+00

B-MATRIX

0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2581784	-0.4131355E-02	0.1253043
0.0000000E+00	0.0000000E+00	0.0000000E+00
-0.4131356E-02	0.3456842	-0.4420982
0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1253043	-0.4420982	3.366339